



WARNING

DO NO MAKE CHANGES TO THIS DOCUMENT
because they can damage WorkBench behaviour:

1.0.0

CORE - LIB

POWER-KI

Library Reference

DESCRIPTION	
PROJECT POWER-KI	CODE XXXXXXXXXXXX
COVER TITLE CORE - LIB	
FIRST TITLE POWER-KI	PAGES 473
SECOND TITLE LIBRARY REFERENCE	PRINT DATE 29/05/20
FILE PWK-MAN-LIB-CORE	SAVED DATE 29/05/20

MANAGED	
ORGANIZATION XPLAB	ENTITY DTC
	MANAGE MXXX.XX

STATUS	
DRAFT	REPLACE //
	REPLACED //

DISTRIBUTION	AVAILABILITY CONFIDENTIAL
--------------	---------------------------

IDENTIFICATION										STORAGE							
SUBJECT	OBJ	CAT	T	TYP	ID	SEQ	VER	LANG		SUBJECT	OBJ	CAT	T	TYP	ID	SEQ	
PXXX.XX	D	XXX	#	XX	XXXXXXXX	XX	1.0.0	EN		PXXX.XX	B	XXX	#	XX	XXXXXXXX	XX	

REVISION									
MAJOR REVISION HISTORY				CREATED/REVISED			APPROVED		
#	NOTE			DATE	BY	NAME	DATE	BY	NAME
0				11/04/11	DTC				

This document contains proprietary information or industrial secrets of

XPLAB s.a.s.

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, translated, transmitted in any form or by any means, without the prior written permission of XPLAB.

©2011-2016 **XPLAB**

XPLAB s.a.s
v.le S.Eufemia, 39
25135 Brescia – ITALY

Tel. +39 030 2350035
Fax. +39 030 2350035

www.xplab.net
www.power-ki.com
www.powerBerry.tech
www.superwire.org
www.xpolyplus.com



Summary

Document Information.....	12
Summary.....	12
Purpose.....	12
Validity.....	12
Relation.....	12
References.....	12
Document Change.....	13
Related Documents.....	14
ERD – External Related Documents.....	14
IRD – Internal Related Documents.....	14
ARD – Attached Related Documents.....	14
Terms and Definition.....	15
Conventions and Symbol.....	16
1 SYS.....	17
1.1 REF.....	18
1.2>NNL.....	19
1.3>NNLv.....	20
1.4 EXIST.....	21
1.5 TYPOF.....	22
1.6 #IF.....	23
1.7 #ELSE.....	24
1.8 #ELSE_OTIF.....	24
1.9 #WHILE.....	25
1.10 #BREAK.....	26
1.11 #SKIP.....	27
1.12 #END.....	28
1.13 ->.....	29
1.14 EXO.....	30
1.15 EXOTHR.....	30
1.16 EXOFLUSH.....	31
1.17 !!.....	32
1.18 !!!.....	32
1.19 \$.....	32
1.20 £.....	33
1.21 ::.....	33
1.22 ?.....	33
1.23 ().....	34
1.24 [].....	34
1.25 { }.....	34
1.26 ~.....	35
1.27 =.....	36
1.28 //.....	37
1.29 MOD.....	38
1.30 ABS.....	39
1.31 /.....	40
1.32 +.....	41
1.33 -.....	41
1.34 +=.....	41
1.35 -=.....	41
1.36 *.....	41
1.37 ==.....	42
1.38 TRUE.....	43
1.39 FALSE.....	44
1.40 RAND.....	45

1.41	UID.....	46
1.42	UCNT.....	47
1.43	DBGLINE.....	48
1.44	EQ.....	49
1.45	!=.....	50
1.46	NE.....	51
1.47	<.....	52
1.48	LT.....	53
1.49	>.....	54
1.50	GT.....	55
1.51	<=.....	56
1.52	LE.....	57
1.53	>=.....	58
1.54	GE.....	59
1.55	&.....	60
1.56	61
1.57	%.....	62
1.58	^.....	63
1.59	SQRT.....	64
1.60	ENP.....	65
1.61	EXP.....	66
1.62	LOG.....	67
1.63	LOG10.....	68
1.64	!.....	69
1.65	<<.....	70
1.66	LSHF.....	71
1.67	>>.....	72
1.68	RSHF.....	73
1.69	<<=.....	74
1.70	++.....	75
1.71	76
1.72	NOT.....	77
1.73	AND.....	78
1.74	OR.....	79
1.75	XOR.....	80
1.76	SWAP.....	81
1.77	MIN.....	82
1.78	MAX.....	83
1.79	LIM.....	84
1.80	IF.....	85
1.81	PI.....	86
1.82	TANH.....	87
1.83	SINH.....	88
1.84	COSH.....	89
1.85	ATAN.....	90
1.86	ATAN2.....	91
1.87	ASIN.....	92
1.88	ACOS.....	93
1.89	TAN.....	94
1.90	SIN.....	95
1.91	COS.....	96
1.92	BITF.....	97
1.93	HEX.....	98
1.94	BIN.....	99
1.95	FLT.....	100
1.96	INT.....	101
1.97	UNS.....	102
1.98	CHAR.....	103
1.99	CHARCOD.....	104



1.100	FRMT.....	105
1.101	NULL.....	106
1.102	TSTX.....	107
1.103	ISNULL.....	108
1.104	ISEMPTY.....	109
1.105	ISTRUE.....	110
1.106	ISERR.....	111
1.107	ISNUM.....	112
1.108	ISFLT.....	113
1.109	QUOS.....	114
1.110	QUOD.....	115
1.111	QUOSE.....	116
1.112	QUODE.....	117
1.113	ESCP.....	118
1.114	CRLF.....	119
1.115	BOM.....	120
1.116	CNSOUT.....	121
1.117	ERROUT.....	122
1.118	ERRSHW.....	123
1.119	TMR.....	124
1.120	CLOCK.....	125
1.121	LIBINF.....	126
1.122	TRASH.....	127
1.123	CMP.....	128
1.124	CHATMSG.....	129
1.125	CHATPUT.....	130
1.126	CHATSHW.....	131
1.127	CHATGET.....	132
1.128	CHATCLR.....	133
1.129	CNS.....	134
1.130	STDOUT.....	135
1.131	STDINP.....	136
1.132	YIELD.....	137
1.133	SLEEP.....	138
1.134	WAITCND.....	139
1.135	SMF.....	140
1.136	DOOR.....	143
1.137	PWKTASK.....	144
1.138	OSEXEC.....	146
1.139	OSSHLL.....	147
1.140	OSSTART.....	148
1.141	CALC.....	149
1.142	LEN.....	150
1.143	NSP.....	151
1.144	FST.....	152
1.145	LST.....	153
1.146	MID.....	154
1.147	SPLT.....	155
1.148	TKNZ.....	156
1.149	TKNZOP.....	157
1.150	CSV.....	158
1.151	CSVTBL.....	159
1.152	SRCH.....	160
1.153	CAT.....	161
1.154	MTCH.....	162
1.155	LWR.....	163
1.156	UPR.....	164
1.157	RTF2TXT.....	165

1.158	PKGPTH.....	166
1.159	PWKPTH.....	167
1.160	EXECMODE.....	168
1.161	ALIASPTH.....	169
1.162	TRIG.....	170
1.163	TRIGSET.....	171
1.164	TRIGPAR.....	172
1.165	LIC.....	173
1.166	THRINF.....	174
1.167	THRSYM.....	175
1.168	THRPRI.....	176
1.169	PTRTYP.....	177
1.170	PTRDUP.....	178
1.171	PTRLIS.....	179
1.172	THIS.....	180
2	SYMB.....	181
2.1	SYMB_NRM.....	182
2.2	SYMB_INDX.....	183
2.3	SYMB_SLCT.....	184
2.4	SYMB_DCD.....	185
2.5	SYMB_RPLC.....	186
2.6	SYMB_BLKSEQ.....	187
2.7	SYMB_BLKBLK.....	188
2.8	SYMB_BLK.....	189
3	DB.....	190
3.1	DB_OPN.....	191
3.2	DB_QRY.....	192
3.3	DB_GET.....	193
4	TREE.....	196
4.1	TREE_OPN.....	197
4.2	TREE_PARSE.....	198
4.3	TREE_SER.....	199
4.4	TREE_SAV.....	200
4.5	TREE_PTH.....	201
4.6	TREE_ITM.....	202
5	DOC.....	204
5.1	DOC_DOC.....	205
5.2	DOC_TPL.....	206
5.3	DOC_OPN.....	207
5.4	DOC_CLS.....	208
5.5	DOC_TBL.....	209
5.6	DOC_TBLDUP.....	210
5.7	DOC_TBRLMV.....	211
5.8	DOC_UF.....	212
5.9	DOC_TVL.....	213
5.10	DOC_TINF.....	214
5.11	DOC_SAV.....	215
6	BUF.....	216
6.1	BUF_NEW.....	217
6.2	BUF_SMF.....	218
6.3	BUF_BUF.....	219
6.4	BUF_VAL.....	220
6.5	BUF_CPY.....	221
6.6	BUF_CAT.....	222
6.7	BUF_CMP.....	223
6.8	BUF_RST.....	224
6.9	BUF_INFO.....	225
6.10	BUF_SYMB.....	226



7	FS	227
7.1	FS_FIND	228
7.2	FS_INFO	229
7.3	FS_LOG	230
7.4	FS_DCWD	231
7.5	FS_DMK	232
7.6	FS_DRM	233
7.7	FS_DRN	234
7.8	FS_FDEL	235
7.9	FS_FCPY	236
7.10	FS_FREN	237
7.11	FS_FOPN	238
7.12	FS_FOPNB	239
7.13	FS_FINP	240
7.14	FS_FOUT	241
7.15	FS_FWRS	242
7.16	FS_FRDS	243
7.17	FS_FWRS_F	244
7.18	FS_FRDS_F	245
7.19	FS_FPOS	246
8	SOK	247
8.1	SOK_NEW	248
8.2	SOK_LKO	249
8.3	SOK_LKC	250
8.4	SOK_LKW	251
8.5	SOK_RCV	253
8.6	SOK_RCVPKT	254
8.7	SOK_INQ	255
8.8	SOK_ADR	256
8.9	SOK_POR	257
8.10	SOK_STS	258
8.11	SOK_SND	259
8.12	SOK_WRS	260
8.13	SOK_RDS	261
8.14	SOK_INF	262
9	IEP	263
9.1	IEP_SRV	264
9.2	IEP_SRVADD	265
9.3	IEP_SRVCMD	266
9.4	IEP_CLI	267
9.5	IEP_CLICON	268
9.6	IEP_CLIDAT	269
9.7	IEP_CLIBUF	270
9.8	IEP_CLIRD	271
9.9	IEP_CLIWR	272
10	FTP	273
10.1	FTP_NEW	274
10.2	FTP_DIR	275
10.3	FTP_GET	276
10.4	FTP_MGET	277
10.5	FTP_PUT	278
10.6	FTP_DEL	279
10.7	FTP_STS	280
10.8	FTP_RECON	281
11	WEB	282
11.1	WEB_NEW	283
11.2	WEB_SET	284

11.3	WEB_CMD.....	285
11.4	WEB_FIL.....	286
11.5	WEB_TXT.....	287
11.6	WEB_BUF.....	288
11.7	WEB_RPL.....	289
12	DT.....	290
12.1	DT_TIME.....	291
12.2	DT_TS.....	292
12.3	DT_TSDEC.....	293
13	/*FREE*/.....	294
14	GUI.....	295
14.1	GUI_OPN.....	296
14.2	GUI_ALV.....	297
14.3	GUI_SND.....	298
14.4	GUI_MOUSE.....	300
14.5	GUI_DUPNEW.....	301
14.6	GUI_DUPDEL.....	302
14.7	GUI_STREAM.....	303
14.8	GUI_FILWR.....	304
14.9	GUI_FILRD.....	305
14.10	GUI_SRV.....	306
14.11	GUI_DLGMSG.....	307
14.12	GUI_DLGOFD.....	308
14.13	GUI_DLGSGFD.....	309
14.14	GUI_DLGFLD.....	310
14.15	GUI_DLGCLR.....	311
14.16	GUI_EXCMMSG.....	312
14.17	GUI_EXCOFD.....	313
14.18	GUI_EXCSFD.....	314
14.19	GUI_EXCFLD.....	315
15	LIS.....	316
15.1	LIS_NEW.....	317
15.2	LIS_CLN.....	318
15.3	LIS_SMF.....	319
15.4	LIS_NUM.....	320
15.5	LIS_USE.....	321
15.6	LIS_PSH.....	322
15.7	LIS_POP.....	323
15.8	LIS_ADD.....	324
15.9	LIS_INS.....	325
15.10	LIS_GET.....	326
15.11	LIS_POS.....	327
15.12	LIS_CLR.....	328
15.13	LIS_SORT.....	329
15.14	LIS_SRCH.....	330
16	INI.....	331
16.1	INI_NEW.....	332
16.2	INI_LOD.....	333
16.3	INI_SAV.....	334
16.4	INI_LBLNUM.....	335
16.5	INI_KEYNUM.....	336
16.6	INI_LBLCOUNT.....	337
16.7	INI_KEYCOUNT.....	338
16.8	INI_LBLLIST.....	339
16.9	INI_KEYLIST.....	340
16.10	INI_KEYGETSET.....	341
16.11	INI_KEYGET.....	342
16.12	INI_KEYSET.....	343



16.13	INI_LBLADD.....	344
16.14	INI_LBLDEL.....	345
16.15	INI_KEYADD.....	346
16.16	INI_KEYDEL.....	347
16.17	INI_CLEAR.....	348
17	NTW.....	349
17.1	NTW_GHBN.....	350
17.2	NTW_GHBA.....	351
17.3	NTW_PING.....	352
18	PKG.....	353
18.1	PKG_OPN.....	354
18.2	PKG_MNFGET.....	355
18.3	PKG_MNFADD.....	356
18.4	PKG_SAV.....	357
18.5	PKG_PTH.....	358
19	KB1.....	359
19.1	PWK function.....	359
19.2	KB1 syntax and function.....	360
19.3	By Example.....	363
19.4	Knowledge Base 1 - internal function.....	364
19.4.1	COUNTLIS.....	365
19.4.2	GETREF.....	365
19.4.3	EXTRACT.....	366
19.4.4	UNLINK.....	366
19.4.6	TBLATT.....	367
19.4.7	IDCHG.....	368
19.4.8	EXISTS.....	368
19.5	KB1_OPN.....	369
19.6	KB1_SMF.....	370
19.7	KB1_SAV.....	371
19.8	KB1_GET.....	372
19.9	KB1_DLG.....	373
19.10	KB1_QRY.....	374
19.11	KB1_EXC.....	375
19.12	KB1_FRM.....	376
19.13	KB1_XNS.....	377
19.14	KB1_XCP.....	378
19.15	KB1_INF.....	379
20	MAIL.....	380
21	TBL.....	381
21.1	TBL_NEW.....	382
21.2	TBL_SMF.....	384
21.3	TBL_CLN.....	385
21.4	TBL_SAV.....	386
21.5	TBL_LOD.....	387
21.6	TBL_CHG.....	388
21.7	TBL_ITM.....	389
21.8	TBL_NAM.....	390
21.9	TBL_INF.....	391
21.10	TBL_EXP.....	393
21.11	TBL_USE.....	394
21.12	TBL_IMP.....	395
21.13	TBL_SRC.....	396
21.14	TBL_EXG.....	397
21.15	TBL_SORT.....	398
22	HTTP.....	399
22.1	HTTP_OPN.....	400

22.2	HTTP_GET.....	401
22.3	HTTP_DEL.....	402
22.4	HTTP_POST.....	403
22.5	HTTP_PUT.....	404
23	WUI.....	405
23.1	WUI_OPN.....	406
23.2	WUI_LOD.....	407
23.3	WUI_SET.....	408
23.4	WUI_HTML.....	410
23.5	WUI_CPTH.....	411
23.6	WUI_SPTH.....	412
24	MTH.....	413
24.1	MTH_PLY.....	414
25	HIO.....	415
25.1	HIO_GPIOnew.....	416
25.2	HIO_GPIOset.....	417
25.3	HIO_GPIO.....	418
25.4	HIO_GPIOpwm.....	419
25.5	HIO_GPIOpulse.....	420
25.6	HIO_PWM.....	421
25.7	HIO_I2Cnew.....	422
25.8	HIO_I2C.....	423
25.9	HIO_SPInew.....	425
25.10	HIO_SPI.....	426
25.11	HIO_UARTnew.....	428
25.12	HIO_UARTset.....	429
25.13	HIO_UART.....	430
26	NPR.....	432
27	OCV.....	433
27.1	OCV_MAT.....	434
27.2	OCV_MAT_INF.....	437
27.3	OCV_MAT_PIX.....	438
27.4	OCV_OP.....	439
27.5	OCV_BKGSUB.....	440
27.6	OCV_BKGRMV.....	442
27.7	OCV_IMREAD.....	443
27.8	OCV_IMWRITE.....	444
27.9	OCV_VCP.....	445
27.10	OCV_VCP_RD.....	446
27.11	OCV_VWR.....	447
27.12	OCV_VWR_WR.....	448
27.13	OCV_QR.....	449
27.14	OCV_RS2.....	450
27.15	OCV_RS2_GET.....	451
28	OPC.....	452
28.1	OPC_NEW.....	453
28.2	OPC_SRV.....	453
28.3	OPC_CLI.....	454
28.4	OPC_RD.....	455
28.5	OPC_GET.....	456
28.6	OPC_WR.....	457
28.7	OPC_MNT.....	458
28.8	OPC_EVT.....	459
28.9	OPC_CALL.....	461
28.10	OPC_DISCOVER.....	462
29	COM.....	463
29.1	COM_NEW.....	464



29.2	COM_SET.....	465
29.3	COM_RCV.....	466
29.4	COM_WAIT.....	467
29.5	COM_SND.....	468
29.6	COM_FNC.....	469
29.7	COM_GMS.....	470
29.8	COM_ERR.....	471
29.9	COM_STS.....	472

Document Information

Summary
Purpose
Validity
Relation
References
[1]

[2]



Document Change

Related Documents

ERD – External Related Documents

N°	DATE	TITLE	SOURCE

IRD – Internal Related Documents

N°	ORG	CODE								DESCRIPTION
		SUBJECT	OBJ	CAT	T	TYP	ID	SEQ	VER	

ARD – Attached Related Documents

N°	ORG	CODE								DESCRIPTION
		SUBJECT	OBJ	CAT	T	TYP	ID	SEQ	VER	



Terms and Definition

Glossary entry	Entry definition
NV	Numerical Value
NNV	NOT Numerical Value
PTR	Pointer
symb	Symbol
slist	List of symb
enum	Enumeration (symb containing values separated by commas)
1bsd	1 based: the first index is 1
0bsd	0 Based; the first index is 0
sep	Separator
op	Operator
fnc	Function
cnst	Constant
IA	Internet Address
N.D.	Not Defined
N.A.	Not Applicable
tbv	to be verified
ERR	NULL or less the 0
rpar	Right Parameter
lpar	Left Parameter
MRF	Multi Return Function

Conventions and Symbol

Text	Description	Example



1

SYS

The following operator/functions are not yet documented:

Fuzzy and Probability: ZNOT, ZAND, ZOR, ZXOR, ZSUM, ZFSUM, PAND, POR, PXOR

1.1 REF

REF		op 1
Description	Instruct PWK to use the parameter as a symbol	
Related		
Remark		

1.REF					
Parameter	Type	Values	Comment	Default	Opt
ELEMENT	symb				
Return					
VAL	symb.				
OnError					
	N.A.				
Example					
<pre>A_01="pippo"; B=ref("A"+"_01"); >>> b=pippo ref("A"+"_01")="papero"; >>> A_01=papero</pre>					



1.2 NNL

NNL		op 1
Description	Never Null - if element is null an empty literal is returned instead	
Related	NNLv (as Function)	
Remark		

1.NNL					
Parameter	Type	Values	Comment	Default	Opt
ELEMENT	symb				
Return					
VAL	symb.				
OnError					
	N.A.				
Example					
A= NULL;					
A= NNL A ==> A==£					

1.3 NNLv

NNLv		fnc
Description	Never Null - if element is null an empty literal or the provided substitute is returned instead	
Related	NNL (as operator)	
Remark		

1.NNLv					
Parameter	Type	Values	Comment	Default	Opt
ELEMENT	symb				
SUBSTITUTE	symb			£	
Return					
VAL	symb.				
OnError					
	N.A.				
Example					
<pre> NNLv("Pippo") >>> "Pippo" NNLv(" ") >>> " " NNLv(NULL) >>> £ (same as "") NNLv(a, "1") >>> if a is null, 1, else value of 'a' </pre>					



1.4 EXIST

EXIST		fnc
Description	Return the address of a literal if Exists	
Related		
Remark	is the same as \$(ref(...)) but if the ELEMENT doesn't exists no error is output on cnserr	

1.EXIST					
Parameter	Type	Values	Comment	Default	Opt
ELEMENT	symb				
Return					
VAL	NV		Address of the element		
OnError					
RES	NULL		Not exists		
Example					
<pre>Main has only 1 attribute: tmp EXIST("tmp@\Main") >>> Positive integer value (attribute pointer, \$tmp@\Main) EXIST("cnd@\Main") >>> NULL</pre>					



1.TYPOF					
Parameter	Type	Values	Comment	Default	Opt
ELEMENT	symp				
Return					
VAL	symp		Type		
OnError					
RES	NULL		Not exists		
Example					
<pre>In the program there is a VAR DATA, containing the MTHD sumClc TYPOF("\DATA\sumClc") >> £MTHD</pre>					



1.6 #IF

#IF		flw
Description	Evaluate a condition if true, the following statements, till #ELSE or #ELSE_OTIF or #END are executed, if false and exists an #ELSE or #ELSE_OTIF, that block of code is executed	
Related	#ELSE, #ELSE_OTIF, #END, #WHILE, #BREAK, #SKIP	
Remark	if more parameter are passed only the last is evaluated for execution, IF alone is the assignment function	

1.#IF					
Parameter	Type	Values	Comment	Default	Opt
CONDITION	symb				
Return					
	N.A.				
OnError					
	N.A.				
Example					
#IF(a>10); CHATput("Greater"); #ELSE; CHATput("Lesser"); #END; CHATput("End") ;					
#IF(a=a+1, a>10); CHATput("Greater");#end;					

1.7 #ELSE

#ELSE		flw
Description	Inside an #IF block, the code between #ELSE and #END is executed, if the condition is false	
Related	#IF, #END, #WHILE, #BREAK, #SKIP	
Remark	IF alone is the assignment function	

1.#ELSE					
Parameter	Type	Values	Comment	Default	Opt
Return					
	N.A.				
OnError					
	N.A.				
Example					
#IF(a>10); CHATput(" Maggiore"); #ELSE; CHATput(" Minore"); #END; put(" Fine")					

1.8 #ELSE_OTIF

#ELSE_OTIF		flw
Description	Inside an #IF block, the code between #ELSE_OTIF and #END is executed, if the condition is false Or, if the #IF condition is TRUE and the #ELSE_OTIF condition is also true.	
Related	#IF, #END, #WHILE, #BREAK, #SKIP	
Remark	OTIF = Or Then IF	

1.#ELSE_OTIF					
Parameter	Type	Values	Comment	Default	Opt
Return					
	N.A.				
OnError					
	N.A.				
Example					
<pre>#if(a); ... chatput(fIF_done); #else_otif(b) ; !! opzionale; ... chatput(fELSE_done); #end; with a=0 : ELSE_done with a=1 e b=0; IF_done with a=1 e b=1: IF_done ELSE_done</pre>					



1.9 #WHILE

#WHILE		flw
Description	the code between #WHILE and #END is executed, if the condition of #WHILE is true	
Related	#IF, #END, #ELSE, #BREAK, #SKIP	
Remark	if more parameter are passed only the last is evaluated for execution,	

1.#WHILE					
Parameter	Type	Values	Comment	Default	Opt
CONDITION	symb				
Return					
	N.A.				
OnError					
	N.A.				
Example					
<pre>a=0; #WHILE(a<10); a=a+1; #END; CHATput(" Fine") ; a=0; #WHILE(a=a+1, a<10); CHATput(">",a);#end;</pre>					

1.10 #BREAK

#BREAK		flw
Description	break a #WHILE	
Related	#IF, #WHILE, #ELSE, #SKIP, #END	
Remark		

1.#BREAK					
Parameter	Type	Values	Comment	Default	Opt
Return					
	N.A.				
OnError					
	N.A.				
Example					
<pre> a=0; #WHILE(a<10); a=a+1; #IF(a==5); #BREAK; #END; #END; </pre>					



1.11 #SKIP

#SKIP		flw
Description	Continue a While without execute the trailing code	
Related	#IF, #WHILE, #ELSE, #END, #BREAK	
Remark		

1.#SKIP					
Parameter	Type	Values	Comment	Default	Opt
Return					
	N.A.				
OnError					
	N.A.				
Example					
<pre>a=0;b=0 #WHILE (a<10); a=a+1; #IF (a==5); #SKIP; #END; b=b+1; #END; >>> a=10, b=9;</pre>					

1.12 #END

#END		flw
Description	mark the end of a block of code inside an #IF, #ELSE, #WHILE	
Related	#IF, #WHILE, #ELSE, #BREAK, #SKIP	
Remark		

1.#END					
Parameter	Type	Values	Comment	Default	Opt
Return					
	N.A.				
OnError					
	N.A.				
Example					
<pre>a=0; #WHILE(a<10); a=a+1; #END; CHATput(" Fine While") ; #IF(a>10); CHATput(" Maggiore"); #ELSE; CHATput(" Minore"); #END; CHATput(" Fine If") ;</pre>					

**1.13 ->**

->		op 18
Description	Right assignmet	
Related		
Remark	pop value from the stack and assign to symbol	

1.->					
Parameter	Type	Values	Comment	Default	Opt
Return					
	N.A.				
OnError					
	N.A.				
Example					
<pre>l=LIS_NEW(1,2,3); LIS_USE(l)->a->b->c; >>> a=3 b=2 c=1 a=LIS_USE(l)->b->c >>> a=1 b=3 c=2 a=LIS_USE(l); >>> not working, 2 pars are unmanaged LIS_USE(l)->b->c; >>> b=3 c=2, 1 elem. remains in stack,but it's cleared on ';' </pre>					

1.14 EXO

EXO		fnc
Description	Inline execution of an EXO	
Related	:: ? EXOTHR	
Remark	Is a MRF	

1.EXO					
Parameter	Type	Values	Comment	Default	Opt
EXO	symb				
PAR	slis		set(::) or get(?) parameter		
Return					
	N.A.				
OnError					
	N.A.				
Example					
<p>SUM is a method that return c=a+b: r=EXO(£\SUM, a:: 5, b:: 6, ?c); ==> internally c=11, so r=11</p> <p>TEST is a method that pass through the values from SET to GET: ra=EXO(£\TEST, a:: 8, b:: 7, c::6, ?a,?b, ?c) -> rc -> rb ; ==> ra=8 rb=7 rc=6</p>					

1.15 EXOTHR

EXOTHR		fnc
Description	Inline start execution of an EXO in a new Thread	
Related	:: ? EXO	
Remark	Is a MRF	

1.EXOTHR					
Parameter	Type	Values	Comment	Default	Opt
EXO	symb				
PAR	slis		set(::)		
Return					
RES	NV	1			
OnError					
ERR	NV	-1			
Example					
<p>r=EXOTHR(£\TEST, a:: 8, b:: 7, c::6) ;</p> <p>A _PAR_ can be defined with the same values of the THREAD item : SYNC ONCE ONCE_FOR_CALLER</p> <p>r=EXOTHR(£\TEST, a:: 8, b:: 7, c::6, _PAR_::£ONCE);</p>					



1.16 EXOFLUSH

EXOFLUSH		fnc
Description	Flush EXO cache	
Related	EXO	
Remark		

1.EXOFLUSH					
Parameter	Type	Values	Comment	Default	Opt
Return					
	N.A.				
OnError					
	N.A.				
Example					
Used modifying the program itself with the TREE functions: EXOFLUSH eliminates the cache of previous executions, to ensure that the last modified version is executed					

1.17 !!

!!		sep
Description	Denote the beginning of a comment. End with semi colon. Colour the comment in GREEN.	
Related	!!!	
Remark	you can inline comment in statement : x= alfa !!example; * beta !!example; ; (remember to close the statement with a semi colon);	

1.18 !!!

!!!		sep
Description	Denote the beginning of a comment. End with semi colon. Colour the comment in BLUE.	
Related	!!	
Remark	you can inline comment in statement : x= alfa !!!example; * beta !!example; ; (remember to close the statement with a semi colon);	

1.19 \$

\$		op 1
Description	Symbol address	
Related	WAITCND, GUI_EXCMMSG	
Remark		

1.\$					
Parameter	Type	Values	Comment	Default	Opt
Rpar	symb				
Return					
address	NV				
OnError					
Example					
<pre>a=\$pippo; Use with REF: \$(REF("pippo")) With attrib a in \Main: \$a >> attrib address \$a@\Main >> same as above and functional, even if marked in yellow as incorrect \$Main >> EXO Main address</pre>					

**1.20 £**

£		op 0
Description	Literal. The following symbol is considered as surrounded by double quote (no white spaces are allowed between character)	
Related		
Remark		

1.£					
Parameter	Type	Values	Comment	Default	Opt
Rpar	cnst				
Return					
Rpar					
OnError					
Example					
BUF_VAL(1, £U32) ; is equal to BUF_VAL(1, "U32") ;					

1.21 ::

::		op0
Description	EXO set parameter at the END of a symbol - the symbol is considered as literal	
Related	EXO ?	
Remark	status:: 5 Note MUST be followed by a value - alone is a separator like comma	

1.22 ?

?		op0
Description	EXO get parameter at the Beginning of a symbol - the symbol is considered as literal	
Related	EXO,::	
Remark	?status	

1.23 ()

	()	sep
Description	Contains function parameters (if more then one separated by comma)	
Example	cat(pippo,pluto,paperino);	
Remark		

1.24 []

	[]	sep
Description	Contains the index of a matrix element.	
Example	Matrice[a,b,c];	
Remark		

1.25 { }

	{ }	op
Description	Indirect reference to a symbol.	
Related		
Remark	each couple of parenthesis represent a level of indirection	

1.{ }					
Parameter	Type	Values	Comment	Default	Opt
Symb	symb		the starting symbol		
Return					
val			the value of a symbol retrieved iterating inside a chain of symbol as indicated by the parenthesis level.		
OnError					
Example					
A='B'; B='foo'; {A} >>> foo					

**1.26** ~

~		op 2
Description	Check a symbol against NULL or empty.	
Related		
Remark		

1.~					
Parameter	Type	Values	Comment	Default	Opt
Rpar	symb				
Return					
val	NV	1	if NOT empty or null		
val	NV	0	if empty or Null		
OnError					
Example					
A = £ ==> ~A >>> 0					
A = " " ==> ~A >>> 0					
A = NULL ==> ~A >>> 0					
A = "Hi" ==> ~A >>> 1					
A = 1 ==> ~A >>> 1					
A = CRLF ==> ~A >>> 1					

1.27 =

=		op 20
Description	Assignment operator	
Related		
Remark		

1.=					
Parameter	Type	Values	Comment	Default	Opt
Lpar	symb				
Rpar	exp				
Return					
OnError					
Example					
a = 1;					
a = b = 1; >>> not working					



1.28 //

//		op 2
Description	Integral division	
Related	MOD	
Remark	division by 0 gives 0	

1.//					
Parameter	Type	Values	Comment	Default	Opt
Lpar	NV				
Rpar	NV				
Return					
VAL	NV		the result of the Lpar integral divided by Rpar		
OnError					
Example					
a = 7 // 2;		>>> a = 3			
7.5 // 3		>>> 2			

1.29 MOD

MOD		op 3
Description	Integral division rest operator	
Related	//	
Remark		

1.MOD					
Parameter	Type	Values	Comment	Default	Opt
Lpar	NV				
Rpar	NV				
Return					
rest	NV		rest of integral division of Lpar by Rpar		
OnError					
Example					
a = 7 MOD 2		>> a = 1			
7.5 MOD 3		>> 1.5			
1.6 MOD 1		>> 0.60000000000000000882 (use to get decimal part of a real)			



1.30 ABS

ABS		op 1
Description	Absolute value	
Related		
Remark		

1.ABS					
Parameter	Type	Values	Comment	Default	Opt
PAR	NV				
Return					
RES	NV		Absolute value of Par		
OnError					
Example					
a = ABS (-15) ; >>> a = 15					

1.31 /

		/		op 3					
Description	Division								
Related									
Remark	division by 0 gives 0								

**1.32 +**

+		op 4
Description	Sum	
Related		
Remark		

1.33 -

-		op 4
Description	Subtraction operator or minus sign.	
Related		
Remark		

1.34 +=

+=		op 4
Description	Atomic Increment	
Related	+= SMF	
Remark	the Left symbol (LV) is incremented by the RV, the assignment is Thread safe	

1.35 -=

-=		op 4
Description	Atomic decrement	
Related	-=, SMF	
Remark	the Left symbol (LV) is decremented by the RV, the assignment is Thread safe	

1.36 *

*		op 3
Description	Multiplication operator.	
Related		
Remark		

1.37 ==

==		op 7
Description	Compare for equality.	
Related	EQ	
Remark	If NNV are involved the comparison is case insensitive and the space at the begin and at the end of the value are ignored. For exact comparison use CMP operator.	

1.==					
Parameter	Type	Values	Comment	Default	Opt
LPAR	symb				
RPAR	symb				
Return					
VAL	NV	1	if equal condition		
VAL	NV	0	if Not equal		
OnError					
Example					
<pre>5 == 4; >>> 0 (False) 5.0 == 5; >>> 1 (True) "Pino" == " Pino "; >>> 1 (True) "Pino" == "Pony"; >>> 0 (False) 5 EQ 4; >>> 0 (False) 5.0 EQ 5; >>> 1 (True) "Pino" EQ "PINO"; >>> 1 (True) "Pino" EQ " Pino "; >>> 1 (True) "Pino" EQ "Pony"; >>> 0 (False) £==£ >> 1 " "=="£ >> 1 (== executes NSP before compare) NULL==£ >> 0 NULL==NULL >> 1 £a==£ >> 0 £a==NULL >> 0</pre>					

**1.38 TRUE**

TRUE		cst
Description	Return 1	
Related	==	
Remark		

1.TRUE					
Parameter	Type	Values	Comment	Default	Opt
Return					
VAL	NV	1			
OnError					
Example					
TRUE	>>> 1				

1.39 FALSE

FALSE		cst
Description	Return 0	
Related	==	
Remark		

1.FALSE					
Parameter	Type	Values	Comment	Default	Opt
Return					
VAL	NV	0			
OnError					
Example					
FALSE >>> 0					

**1.40 RAND**

RAND		fnc
Description	Random generated number (FLT) between 0 and 10000	
Related		
Remark		

1.RAND					
Parameter	Type	Values	Comment	Default	Opt
Return					
VAL	NV		Random number		
OnError					
Example					

1.41 UID

UID		fnc
Description	A unic id built chaining: clock_cnt	
Related		
Remark	It is granted to be unic inside the application and Thread safe	

1.UID					
Parameter	Type	Values	Comment	Default	Opt
Return					
VAL	symb		Random number		
OnError					
Example					
UID() ==> "1455110042_1"					



1.42 UCNT

UCNT		fnc
Description	A unic counter incremented at each call	
Related		
Remark	It is granted to be unic inside the application and Thread safe	

1.UCNT					
Parameter	Type	Values	Comment	Default	Opt
Return					
VAL	symb		Random number		
OnError					
Example					
UCNT () >>> 2					
UCNT >>> 3					
UCNT >>> 4					

1.43 DBGLINE

DBGLINE		cst
Description	Return the current code line	
Related		
Remark		

1.DBGLINE					
Parameter	Type	Values	Comment	Default	Opt
Return					
VAL	NV		current code line number		
OnError					
Example					
dbgline >>> 1200 chatput("[ERR]error at line ",dbgline) >>> chat : [ERR]error at line 1200					



1.44 EQ

EQ		op 7
Description	Compare for equality	
Related	==	
Remark	If NNV are involved the comparison is case insensitive and the space at the begin and at the end of the value are ignored. For exact comparison use CMP operator.	

1.EQ					
Parameter	Type	Values	Comment	Default	Opt
LPAR	symb				
RPAR	symb				
Return					
VAL	NV	1	if equal condition		
VAL	NV	0	if Not equal		
OnError					
Example					
5 == 4; >>> 0 (False)					
5.0 == 5; >>> 1 (True)					
"Pino" == " Pino "; >>> 1 (True)					
"Pino" == "Pony"; >>> 0 (False)					
5 EQ 4; >>> 0 (False)					
5.0 EQ 5; >>> 1 (True)					
"Pino" EQ "PINO"; >>> 1 (True)					
"Pino" EQ " Pino "; >>> 1 (True)					
"Pino" EQ "Pony"; >>> 0 (False)					

1.45 !=

!=		op 2
Description	Compare for inequality.	
Related	NE , <> DEPRECATED	
Remark	If NNV are involved the comparison is case insensitive and the space at the begin and at the end of the value are ignored. For exact comparison use CMP operator.	

1.!=					
Parameter	Type	Values	Comment	Default	Opt
LPAR	symb				
RPAR	symb				
Return					
VAL	NV	1	if LPAR NOT equal toRPAR		
VAL	NV	0	if LPAR equal to RPAR		
OnError					
Example					
see EQ					



1.46 NE

NE		op 7
Description	Compare for inequality.	
Related	!=	
Remark	If NNV are involved the comparison is case insensitive and the space at the begin and at the end of the value are ignored. For exact comparison use CMP operator.	

1.NE					
Parameter	Type	Values	Comment	Default	Opt
LPAR	symb				
RPAR	symb				
Return					
VAL	NV	1	if LPAR NOT equal toRPAR		
VAL	NV	0	if LPAR equal to RPAR		
OnError					
Example					
see EQ					

1.47 <

<		op 6
Description	compare for Less then	
Related	LT	
Remark		

1.<					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV	1	if LPAR is less then RPAR		
VAL	NV	0	if LPAR lis NOT Less then RPAR		
OnError					
Example					



1.48 LT

LT		op 6
Description	compare for Less then	
Related	<	
Remark		

1.LT					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV	1	if LPAR is less then RPAR		
VAL	NV	0	if LPAR lis NOT Less then RPAR		
OnError					
Example					

1.49 >

>		op 6
Description	Compare for Greater Then	
Related	GT	
Remark		

1.>					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV	1	if LPAR is Greater then RPAR		
VAL	NV	0	if LPAR is NOT Greater then RPAR		
OnError					
Example					

**1.50 GT**

GT		op 6
Description	Compare for Greater Then	
Related	<	
Remark		

1.GT					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV	1	if LPAR is Greater then RPAR		
VAL	NV	0	if LPAR lis NOT Greater then RPAR		
OnError					
Example					

1.51 <=

<=		op 6
Description	compare for Less or equal then	
Related	LE	
Remark		

1.<=					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV	1	if LPAR is Less Or Equal then RPAR		
VAL	NV	0	if LPAR lIs NOT Less Or Equal then RPAR		
OnError					
Example					
0<=1	>> 1				
5<=7	>> 1				
a<=0	>> 1	(symb is valued 0)			
£<=0	>> 1	(£ is a symb)			
NULL<=0	>> 1	(NULL is valued 0)			



1.52 LE

LE		op 6
Description	compare for Less or equal then	
Related	<=	
Remark		

1.LE					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV	1	if LPAR is Less Or Equal then RPAR		
VAL	NV	0	if LPAR lis NOT Less Or Equal then RPAR		
OnError					
Example					

1.53 >=

>=		op 6
Description	compare for Greater or Equal then	
Related	GT	
Remark		

1.>=					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV	1	if LPAR is Greater or Equal then RPAR		
VAL	NV	0	if LPAR lis NOT Greater or Equal then RPAR		
OnError					
Example					



1.54 GE

GE		op 6
Description	compare for Greater or Equal then	
Related	>=	
Remark		

1.GE					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV	1	if LPAR is Greater or Equal then RPAR		
VAL	NV	0	if LPAR lis NOT Greater or Equal then RPAR		
OnError					
Example					

1.55 &

&		op 8
Description	binary AND	
Related		
Remark		

1.&					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV		the binary AND of LPAR and RPAR		
OnError					
Example					
a=0b1010; b=0b1100; a & b; >>> 0b1000					



1.56

|

		op 10
Description	binary OR	
Related		
Remark		

1.					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV		the binary OR of LPAR and RPAR		
OnError					
Example					
a=0b1010; b=0b1100; a b; >>> 0b1110					

1.57 %

%		op 9
Description	binary XOR	
Related		
Remark		

1.%					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV		the binary XOR of LPAR and RPAR		
OnError					
Example					
a=0b1010; b=0b1100; a % b; >>> 0b0110					



1.58

^

^		op 1
Description	Power	
Related		
Remark		

1.^					
Parameter	Type	Values	Comment	Default	Opt
LPAR	NV				
RPAR	NV				
Return					
VAL	NV		LPAR raised to RPAR		
OnError					
Example					
a=2; b=a^3; >>> b=8					

1.59 Sqrt

Sqrt		fnc
Description	Square Root	
Related		
Remark	If par1 is negative, Sqrt returns 0	

1.Sqrt					
Parameter	Type	Values	Comment	Default	Opt
Par1	NV		NOT negative value		
Return					
VAL	NV		square root of Par1		
OnError					
Example					
sqrt (4)	>>>	2			



1.60 ENP

ENP		cst
Description	return Neperian value	
Related	EXP,LOG	
Remark	Neperian value (2.7...)	

1.61 EXP

EXP		fnc
Description	Calculate neperian exponent	
Related	ENP,LOG	
Remark		

1.EXP					
Parameter	Type	Values	Comment	Default	Opt
PAR1	NV				
Return					
VAL			neperian exponent of Par1		
OnError					
Example					
EXP(1)	>> 2.7182...				



1.62 LOG

LOG		fnc
Description	Calculate neperian logarithm	
Related	ENP,EXP	
Remark		

1.LOG					
Parameter	Type	Values	Comment	Default	Opt
PAR1	NV				
Return					
VAL	NV		neperian Logarithm of Par1		
OnError					
Example					
LOG (ENP) >> 1					
LOG (ENP^3) >> 3					

1.63 LOG10

LOG10		fnc
Description	Calculate 10 based logarithm	
Related	ENP, EXP, LOG	
Remark		

1.LOG10					
Parameter	Type	Values	Comment	Default	Opt
PAR1	NV				
Return					
VAL	NV		10 based logarithm of PAR1		
OnError					
Example					
LOG10(1000) >>> 3					

**1.64 !**

!		op 2
Description	NOT operator	
Related		
Remark		

1.!					
Parameter	Type	Values	Comment	Default	Opt
Rpar	NV				
Return					
VAL	NV	1	if Rpar == 0		
VAL	NV	0	if Rpar <> 0		
OnError					
Example					

1.65 <<

<<		OP 5
Description	Left shift	
Related	LSHF	
Remark		

1.<<					
Parameter	Type	Values	Comment	Default	Opt
PAR1	symb		behaviour change if PAR1 contains NV or NNV		
count	NV				
Return					
VAL	NV		PAR1 shifted left as many BIT indicated by count		
VAL	NNV		PAR1 shifted left as many CHR indicated by count		
OnError					
Example					
NV: a=1; b = a << 2; >>> b = 4 c = a LSHF 2; >>> c = 4 NNV: a="Pippo"; b = a << 2; >>> b = "ppo" c = a LSHF 2; >>> c = "ppo"					



1.66 LSHF

LSHF		OP 5
Description	Left shift	
Related	<<	
Remark		

1.LSHF					
Parameter	Type	Values	Comment	Default	Opt
PAR1	symb		behaviour change if PAR1 contains NV or NNV		
count	NV				
Return					
VAL	NV		PAR1 shifted left as many BIT indicated by count		
VAL	NNV		PAR1 shifted left as many CHR indicated by count		
OnError					
Example					
NV: a=1; b = a << 2; >>> b = 4 c = a LSHF 2; >>> c = 4					
NNV: a="Pippo"; b = a << 2; >>> b = "ppo" c = a LSHF 2; >>> c = "ppo"					

1.67 >>

>>		op 5
Description	Right Shift	
Related	RSHF	
Remark		

1.>>					
Parameter	Type	Values	Comment	Default	Opt
PAR1	symb		behaviour change if PAR1 contains NV or NNV		
count	NV				
Return					
VAL	NV		PAR1 shifted right as many BIT indicated by count		
VAL	NNV		PAR1 shifted right as many CHR indicated by count		
OnError					
Example					
NV: a=8; b = a >> 2; >>> b = 2 c = a RSHF 2; >>> c = 2 1>>2 >>> 0					
NNV: a="Pippo"; b = a >> 2; >>> b = "Pip" c = a RSHF 2; >>> c = "Pip"					



1.68 RSHF

RSHF		op 5
Description	Right Shift	
Related	>>	
Remark		

1.RSHF					
Parameter	Type	Values	Comment	Default	Opt
PAR1	symb		behaviour change if PAR1 contains NV or NNV		
count	NV				
Return					
VAL	NV		PAR1 shifted right as many BIT indicated by count		
VAL	NNV		PAR1 shifted right as many CHR indicated by count		
OnError					
Example					
NV:					
a=8;					
b = a >> 2; >>> b = 2					
c = a RSHF 2; >>> c = 2					
NNV:					
a="Pippo";					
b = a >> 2; >>> b = "Pip"					
c = a RSHF 2; >>> c = "Pip"					



1.69 <<=

<<=		op 14
Description	Concatenate symbol	
Related	.. , ++	
Remark	same as ++	

1.<=<					
Parameter	Type	Values	Comment	Default	Opt
Lpar	symb				
Rpar	symb				
Return					
result	symb		the concatenation of Lpar and Rpar		
OnError					
Example					
a="tre"; b="no"; c=a<=<b; c >>> "treno"					

**1.70 ++**

++		op 14
Description	Concatenate symbol	
Related	.. , <<=	
Remark	same as <<=	

1.++					
Parameter	Type	Values	Comment	Default	Opt
Lpar	symb				
Rpar	symb				
Return					
result	symb		the concatenation of Lpar and Rpar		
OnError					
Example					
a="tre"; b="no"; c=a++b; c >>> "treno"					
a++f >>> a (do nothing)					
a++NULL >>> a (do nothing)					

1.71 ..

..		op 14
Description	Concatenate symbol with a space between	
Related	<<=	
Remark		

1...					
Parameter	Type	Values	Comment	Default	Opt
Lpar	symb				
Rpar	symb				
Return					
result	symb		the concatenation of Lpar and Rpar		
OnError					
Example					
<pre>a="tre"; b="no"; c=a .. b; c >>> "tre no"</pre>					



1.72 NOT

NOT		op 2
Description	LOGICAL NOT	
Related	!	
Remark		

1.NOT					
Parameter	Type	Values	Comment	Default	Opt
Rpar	symb				
Return					
symb	NV	0	if Rpar is 0 or NULL or empty		
symb	NV	1	if Rpar is NOT (0 or NULL or empty)		
OnError					
Example					
NV:					
a=1;					
!a		>>>	0		
NNV:					
!("Pino" <> "PINO");		>>>	1	(<> is case insensitive)	

1.73 AND

AND		op 11
Description	LOGICAL AND	
Related		
Remark		

1.AND					
Parameter	Type	Values	Comment	Default	Opt
Lpar	symb				
Rpar	symb				
Return					
rslt	NV	0	false		
rslt	NV	1	true		
OnError					
Example					
NV: a=1; b=0; a AND b; >>> 0 (False)					
NNV: a=1; ("Pino" == "PINO") AND a; >>> 1 (True)					



OR

OR		op 13
Description	LOGICAL OR	
Related		
Remark		

1.OR					
Parameter	Type	Values	Comment	Default	Opt
Rpar	symb				
Lpar	symb				
Return					
rslt	NV	0	false		
rslt	NV	1	true		
OnError					
Example					
<p>NV:</p> <pre>a=1; b=0; a OR b; >>> 1 (True)</pre> <p>NNV:</p> <pre>a=0; ("Pino" <> "PINO") OR a; >>> 0 (False)</pre>					

1.75 XOR

XOR		op 12
Description	LOGICAL XOR	
Related		
Remark		

1.XOR					
Parameter	Type	Values	Comment	Default	Opt
Lpar	symb				
Rpar	symb				
Return					
rslt	NV	0	false		
rslt	NV	1	true		
OnError					
Example					
NV: a=1; b=0; a XOR b; >>> 1 (True)					
NNV: a=1; ("Pino" == "PINO") XOR a; >>> 0 (False)					



1.76 SWAP

SWAP		fnc
Description	swap a symb BYTE,WORD, DWORD	
Related		
Remark		

1.SWAP					
Parameter	Type	Values	Comment	Default	Opt
PAR1	symb		symb to swap		
swapType	symb	£BYTE;£WORD; £DWORD			
ReturnTyp		£U8;£C8;£I8;£U16; £C16;£I16;£U32; £I32;£F32;	if not specified I returned as HEX		yes
Return					
VAL	NV		Swapped symbol		
OnError					
res	symb	NULL			
Example					
SWAP (55, £BYTE) >> 0x73 return result in hex format					
SWAP (0x12, £BYTE) >> 0x21 exchange the first 4 bits with the seconds					
SWAP (0x1234, £BYTE) >> 0x43 discard part exceding 1 byte					
SWAP (0xaabb, £WORD) >> 0xbbaa exchange the first 8 bytes with the seconds					
SWAP (0xaabbccdd, £WORD) >> 0xddcc discard part exceding 1 word					
SWAP (0xaabbccdd, £DWORD) >> 0xddccbbaa					
SWAP (0x12345678, £DWORD) >> 0x78563412					

1.77 MIN

MIN		fnc
Description	Minimum value of a list of values	
Related	MAX	
Remark		

1.MIN					
Parameter	Type	Values	Comment	Default	Opt
...	sympb		slis of values		
Return					
min	NV		the minimum value of the list		
OnError					
Example					
<pre>a=1;b=4 MIN(a,5,b); >>> 1</pre>					



1.78 MAX

MAX		fnc
Description	maximum value of a list of values	
Related	MIN	
Remark		

1.MAX					
Parameter	Type	Values	Comment	Default	Opt
...	symb		slis of values		
Return					
max	NV		the maximum value of the list		
OnError					
Example					
a=1;b=4 MAX(a,5,b); >>> 5					

1.79 LIM

LIM		fnc
Description	Constraint a value between limit	
Related	MIN, MAX	
Remark	For MIN and MAX a NULL value prevent the relative comparison	

1.LIM					
Parameter	Type	Values	Comment	Default	Opt
VAL	NV				
MIN	NV				
MAX	NV				YES
Return					
VAL	NV		MIN if VAL < Min , MAX if VAL > MAX		
OnError					
Example					
LIM(0, 0.5, 1)		>> 0.5			
LIM(2, 0.5, 1)		>> 1			
LIM(5, 2, 7)		>> 5			



1.80 IF

IF		fnc
Description	ternary IF assignment	
Related		
Remark	if val1 or and if val2 are statement they are always executed	

1.IF					
Parameter	Type	Values	Comment	Default	Opt
cond	NV				
val1	symb				
val2	symb				
Return					
val1	symb		if cond is true		
val2	symb		if cond is false		
OnError					
Example					
<pre>a=1; IF(a>0,"True","False"); >>> "True" IF(0,12,"Always Zero"); >>> "Always Zero"</pre>					

1.81 **PI**

PI		cst
Description	Return greeck pi value 3.1412...)	
Related		
Remark		



1.82 **TANH**

TANH		fnc
Description	Hyperbolic Tangent	
Related		
Remark		

1.TANH					
Parameter	Type	Values	Comment	Default	Opt
angle	NV		The angle in radiant		
Return					
tanh	NV		The tanh of angle		
OnError					
Example					

1.83 SINH

SINH		fnc
Description	Hyperbolic Sin	
Related		
Remark		

1.SINH					
Parameter	Type	Values	Comment	Default	Opt
angle	NV		in radiant		
Return					
sinh	NV		The sinh of angle		
OnError					
Example					



1.84 COSH

COSH		fnc
Description	Hyperbolic Cos	
Related		
Remark		

1.COSH					
Parameter	Type	Values	Comment	Default	Opt
angle	NV		angle in radiant		
Return					
cosh	NV		The hyperbolic cos of angle		
OnError					
Example					

1.85 ATAN

ATAN		fnc
Description	Arc of tangent	
Related		
Remark		

1.ATAN					
Parameter	Type	Values	Comment	Default	Opt
tangent	NV				
Return					
angle	NV		in radiant		
OnError					
Example					



1.86 ATAN2

ATAN2		fnc
Description	Returns the principal value of the arc tangent of y/x , expressed in radians.	
Related		
Remark		

1.ATAN2					
Parameter	Type	Values	Comment	Default	Opt
Y	NV				
X	NV				
Return					
angle	NV		in radiant		
OnError					
Example					

1.87 ASIN

ASIN		fnc
Description	Arc of sinus	
Related		
Remark		

1.ASIN					
Parameter	Type	Values	Comment	Default	Opt
sin	NV				
Return					
angle	NV		in radiant		
OnError					
Example					



1.88 ACOS

ACOS		fnc
Description	Arc of cosinus	
Related		
Remark		

1.ACOS					
Parameter	Type	Values	Comment	Default	Opt
cosinus	NV				
Return					
angle	NV		in radiant		
OnError					
Example					

1.89 TAN

TAN		fnc
Description	Tangent	
Related		
Remark		

1.TAN					
Parameter	Type	Values	Comment	Default	Opt
angle	NV		in radiant		
Return					
tangent	NV				
OnError					
Example					



1.90 SIN

SIN		fnc
Description	Sinus of angle	
Related		
Remark		

1.SIN					
Parameter	Type	Values	Comment	Default	Opt
angle	nv		in radiant		
Return					
sin	NV				
OnError					
Example					

1.91 COS

COS		fnc
Description	the co-sinus of angle	
Related		
Remark		

1.COS					
Parameter	Type	Values	Comment	Default	Opt
angle	NV		in radiant		
Return					
cos	NV				
OnError					
Example					



1.92 BITF

BITF		fnc
Description	set or get a bit field	
Related		
Remark	Numbering of bit from right to left, from 1 to 32	

1.BITF					
Parameter	Type	Values	Comment	Default	Opt
VALUE	NV				
OFFESET	NV		Field Offset 1-32		
LEN	NV		Field Len in bit	1	YES
FIELDVAL	NV		(set)Field new val		YES
VALUE_SIZE	NV		In bit the size (max32)	32	YES
Return					
rsIt	NV		(GET)field value . (SET) changed value		
OnError					
ERR	symb	NULL			
Example					
a=0B1010; b=BITF(a,3,2); >>> b=0B01 (GET) c=BITF(a,3,2,0b10); >>> b=0B1100 (SET)					

1.93 HEX

HEX		fnc
Description	Convert NV in hexadecimal representation prefixed by OX.	
Related		
Remark		

1.HEX					
Parameter	Type	Values	Comment	Default	Opt
PAR	NV				
Return					
rslt	NV		hexadecimal representation of PAR prefixed by OX		
OnError					
Example					
<pre> a=10; b=HEX (a) ; >>> b=0XA c=HEX (48) ; >>> c=0X30 </pre>					



1.94 BIN

BIN		fnc
Description	Convert NV in binary representation prefixed by OB.	
Related		
Remark		

1.BIN					
Parameter	Type	Values	Comment	Default	Opt
PAR	NV				
Return					
rsIt	NV		binary representation of PAR prefixed by OB.		
OnError					
Example					
a=10; b=BIN(a); >>> b=0B1010 c=BIN(15); >>> c=0B1111					

1.95 FLT

FLT		fnc
Description	Convert NV in float with a specified number of fractional digit.	
Related		
Remark	FLT is different from TRUNCATE in that it approximates the closest value both by excess and by defect	

1.FLT					
Parameter	Type	Values	Comment	Default	Opt
VAL	NV				
Precision	NV		if not specified the maximum precision is used		YES
Return					
VAL	NV				
OnError					
Example					
<pre> a=10.546; b=FLT(a,2); >>> b=10.55 c=FLT(15,1); >>> c=15.0 FLT(15.59, 1) >>> 15.6 </pre>					



1.96 INT

INT		fnc
Description	Convert NV to integer	
Related		
Remark		

1.INT					
Parameter	Type	Values	Comment	Default	Opt
PAR	NV				
BITlen	NV	0;8;12;16;20;24;28	Len in bit for sign propagation	0=32	YES
Return					
VAL	NV		the integer value of PAR (0 if PAR in NNV)		
OnError					
Example					
INT(-7.6)		>> 8			
INT(1.4)		>> 1			
INT(1.5)		>> 2			

1.97 UNS

UNS		fnc
Description	Convert NV to unsigned	
Related		
Remark		

1.UNS					
Parameter	Type	Values	Comment	Default	Opt
PAR	NV				
Return					
VAL	NV		the unsigned value of PAR 0 if PAR in NNV		
OnError					
Example					
UNS(-7.6)		>> 4294967288			
UNS(1.4)		>> 1			
UNS(1.5)		>> 2			



1.98 CHAR

CHAR		fnc
Description	Encode a list of NV as Char and return a symbol of concatenated values	
Related		
Remark		

1.CHAR					
Parameter	Type	Values	Comment	Default	Opt
charCode	NV		slist of NV (0-65535)		
Return					
rsIt	symb		the result of conversion and concatenation		
OnError					
Example					
CHAR (48,49,80); >>> "01P"					

1.99 CHARCOD

CHARCOD		fnc
Description	Return the NV of a symb character	
Related		
Remark		

1.CHARCOD					
Parameter	Type	Values	Comment	Default	Opt
CHAR	symb		The first char is evaluated		
Return					
CODE	NV		The code		
OnError					
Example					
<pre>CHARCOD("Ab") >>> 65 (Translate only the first if specified a set) CHARCOD("Ø") >>> 216 CHARCOD("~") >>> 126</pre>					



1.100 FRMT

FRMT		fnc
Description	Format a string with the behavior of C printf	
Related		
Remark	slist SHOULD match the type requested by format	

1.FRMT					
Parameter	Type	Values	Comment	Default	Opt
format	symb		formed with C printf notation		
...	slist		list of symbol		
Return					
rslt	symb		A symb formatted		
OnError					
Example					
<pre>a=12; FRMT("Valore %d",a); >>> "Valore 12" FRMT("Characters: %c %c \n", 'a', 65); >>> Characters: a A FRMT("Decimals: %d %ld\n", 1977, 650000); >>> Decimals: 1977 650000 FRMT("Preceding with blanks: %10d \n", 1977); >>> Preceding with blanks: 1977 FRMT("Preceding with zeros: %010d \n", 1977); >>> Preceding with zeros: 0000001977 FRMT("Some different radixes: %d %x %o %#x %#o \n",100,100,100,100,100); >>> Some different radixes: 100 64 144 0x64 0144 FRMT("floats: %4.2f %+.0e %E \n", 3.1416, 3.1416, 3.1416); >>> floats: 3.14 +3e+000 3.141600E+000 FRMT("Width trick: %*d \n", 5, 10); >>> Width trick: 10 FRMT("%s \n", "A string"); >>> A string FRMT("%04X", 11) >>> 000B</pre>					

1.101 NULL

NULL		cst
Description	return a NULL symbol	
Related		
Remark		



1.102 TSTX

TSTX		fnc
Description	Test a condition and return left (TRUE) or right(FALSE) value	
Related		
Remark		

1.TSTX					
Parameter	Type	Values	Comment	Default	Opt
CONDITION	symb				
VALUES	enum			TRUE,FALSE	YES
SEPARATOR	SYMB			,	YES
Return					
VAL	symb				
OnError					
Example					
<pre>b=1; c=TSTx(b); >> c: TRUE c=TSTx(b, "pippo,Paperino"); >> c: pippo TSTx(1, "testa croce", " ") >> ftesta TSTx(0, "testa croce", " ") >> fcroce TSTx(0, "1;2;3", ";") >> "2;3" (it considers only the first separator)</pre>					

1.103 ISNULL

ISNULL		fnc
Description	Verify if parameter is NULL	
Related	ISEMPTY, ISTRUE, ~	
Remark		

1.ISNULL					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
VAL	NV	1	if PAR is NULL		
VAL	NV	0	if PAR is NOT NULL		
OnError					
Example					
ISNULL (£ciao) >> 0					
ISNULL (" ") >> 0					
ISNULL (£) >> 0					
ISNULL (NULL) >> 1					



1.104 ISEEMPTY

ISEEMPTY		fnc
Description	Verify if parameter is EMPTY (no contents or all spaces)	
Related	ISNULL, ISTRUE, ~	
Remark		

1.ISEEMPTY					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
VAL	NV	1	PAR is EMPTY		
VAL	NV	0	PAR is NOT EMPTY		
OnError					
Example					
ISEEMPTY(" a")		>>> 0			
ISEEMPTY(\$pippo)		>>> 0			
ISEEMPTY(CRLF)		>>> 0			
ISEEMPTY(" ")		>>> 1			
ISEEMPTY(NULL)		>>> 0 (???)			
ISEEMPTY(\$)		>>> 1			

1.105 ISTRUE

ISTRUE		fnc
Description	Verify if parameter is TRUE	
Related	ISNULL, IEMPTY, ~	
Remark	NV are true if not NULL or 0; NNV are true if not NULL or EMPTY	

1.ISTRUE					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
VAL	NV	1	PAR is TRUE		
VAL	NV	0	PAR is FALSE		
OnError					
Example					
ISTRUE(0) ==> 0 ISTRUE(NULL) ==> 0 ISTRUE(" ") ==> 0 ISTRUE(10.5) ==> 1 ISTRUE(-1) ==> 1 ISTRUE(\$A) ==> 1					



1.106 ISERR

ISERR		fnc
Description	Verify if PAR is ERROR (NULL or <0)	
Related		
Remark		

1.ISERR					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
VAL	NV	1	if PAR is ERR		
VAL	NV	0	if PAR is NOT ERR		
OnError					
Example					
ISERR (1) >>> 0					
ISERR (0) >>> 0					
ISERR (-1) >>> 1					
ISERR (£ciao) >>> 0					
ISERR (£) >>> 1					
ISERR (" ") >>> 1					

1.107 ISNUM

ISNUM		fnc
Description	test par to verify if it is a number	
Related		
Remark		

1.ISNUM					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
VAL	NV	1	if is a number		
VAL	NV	0	if is NOT a number		
OnError					
Example					
<pre> a="Pippo"; ISNUM(a) >>> 0 a="10"; ISNUM(a) >>> 1 a=10; ISNUM(a) >>> 1 a=2.5; ISNUM(a) >>> 1 </pre>					



1.108 ISFLT

ISFLT		fnc
Description	test par to verify if it is a float	
Related		
Remark		

1.ISFLT					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
VAL	NV	1	if is a FOAT		
VAL	NV	0	if is NOT a FLOAT		
OnError					
Example					
a="Pippo";					
ISFLT(a)	>>>	0			
a="1.5";					
ISFLT(a)	>>>	1			
a=10;					
ISFLT(a)	>>>	0			
a=2.5;					
ISFLT(a)	>>>	1			
ISFLT(2.0)	>>>	1			

1.109 QUOS

QUOS		fnc
Description	Single quote	
Related	QUOD,QUOSE,QUODE,ESCP	
Remark		

1.QUOS					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
rslt	symb		return PAR surrounded by single quote		
OnError					
Example					
<pre>a=10.546; b=QUOS (a) ; >>> b=<'10.546'></pre>					



1.110 QUOD

QUOD		fnc
Description	Double quote	
Related	QUOS,QUOSE,QUODE,ESCP	
Remark		

1.QUOD					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
rsIt	symb		return PAR surrounded by double quote		
OnError					
Example					
a=10.546; b=QUOD (a) ; >>> b=<"10.546">					

1.111 QUOSE

QUOSE		fnc
Description	Single quote and Escape doubling the single quote	
Related	QUOS,QUOD,QUODE,ESCP	
Remark		

1.QUOSE					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
rslt	symb		return PAR surrounded by single quote		
OnError					
Example					
<pre>a=" val='al' "; b=QUOSE(a); >>> b=< val ='al' ' ></pre>					



1.112 QUODE

QUODE		fnc
Description	Double quote and Escape doubling the double quote	
Related	QUOS,QUOD,QUOSE,ESCP	
Remark		

1.QUODE					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
rsIt	symb		return PAR surrounded by double quote		
OnError					
Example					
<pre>a=" val="al" "; b=QUODE(a); >>> b=< val =""al"" ></pre>					

1.113 ESCP

ESCP		fnc
Description	Escape a character in a symbol	
Related	QUOS,QUOD,QUOSE,ESCP	
Remark		

1.ESCP					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		The symbol to escape		
CHRtoEscape	symb		A single character		
ESCAPEwith	symb		the escape symbol		
Return					
rsIt	symb		return PAR escaped		
OnError					
err	symb	NULL			
Example					
<pre>a=" val='al' "; b=ESCP(a,"'", "'"); >>> b=" val ='al\' "</pre>					



1.114 CRLF

CRLF		cst
Description	return a symbol containing Carriage return and line feed (0x0d,0x0a)	
Related		
Remark		

1.115 BOM

BOM		cst
Description	return a symbol containing Byte Order Mark (0xFEFF) for unicode text	
Related		
Remark	0xFEFF	



1.116 CNSOUT

CNSOUT		fnc
Description	Output on the standard output device the concatenation of the slist of symbol	
Related		
Remark		

1.CNSOUT					
Parameter	Type	Values	Comment	Default	Opt
...	slist		list of symbol		
Return					
OnError					
Example					

1.117 ERROUT

ERROUT		fnc
Description	Output on the standard error device the concatenation of the slist of symbol	
Related		
Remark		

1.ERROUT					
Parameter	Type	Values	Comment	Default	Opt
...	slist		list of symbol		
Return					
OnError					
Example					

**1.118 ERRSHW**

ERRSHW		fnc
Description	Return in a symbol system error output by the system	
Related		
Remark		

1.ERRSHW					
Parameter	Type	Values	Comment	Default	Opt
Return					
TEXT	symb		System error writed in err cons		
OnError					
Example					

1.119

TMR		fnc
Description	read the system timer or the Thread timer or the Hres timer (FLOAT ms.us)	
Related		
Remark		

1.TMR					
Parameter	Type	Values	Comment	Default	Opt
PAR	symbol	£SYS;£THREAD;£HRES(in FLT ms.us);£MICRO(in micro the actual time + TMP if done); £EXPIRED(check a micro time against the TMP); £MICRODIF(AT - TMP)		SYS	YES
TMP	NV		in MICRO		YES
Return					
timer	NV		Ms since program started		
EXPIRED	NV		0 NOT - 1 YES		
OnError					
Example					
HRES time in FLT ms.us					
MICRO return the actual time (AT) in micro					
TMR		>> milliseconds since the program started			
TMR (£SYS)		>> as above			
TMR (£THREAD)		>> milliseconds since the thread started			
TMR (£HRES)		>> float milliseconds dot 3 digit microseconds			
TMR (£MICRO)		>> microseconds			
TMR (£EXPIRED,time)		>> 1 if time(ms) is expired			
TMR (£MICRODIF,time)		>> difference between current timer and time			



1.120 CLOCK

CLOCK		fnc
Description	Read system clock.	
Related		
Remark		

1.CLOCK					
Parameter	Type	Values	Comment	Default	Opt
Return					
time	NV		Return second since 00:00 of January 1 st 1970		
OnError					
Example					

1.121 LIBINF

LIBINF		fnc
Description	Library information	
Related		
Remark		

1.LIBINF					
Parameter	Type	Values	Comment	Default	Opt
PAR	symp		library name		yes
Return					
functions	enum		of functions, If called with PAR		
libraries	enum		of libraries, if called without par		
OnError					
Example					
LIBINF(£DT) >>> "DT_TS,DT_TSDEC,DT_TIME"					



1.122 TRASH

TRASH		fnc
Description	delete system pointer and set to NULL the attribute	
Related		
Remark		

1.TRASH					
Parameter	Type	Values	Comment	Default	Opt
...	slist		can be system pointer or library name. In this last case, all allocated system pointer of the library are deleted.		
Return					
OnError					
Example					
<pre>ptr=LIS_NEW; !!now ptr has a value; TRASH(ptr); !!now ptr is NULL;</pre>					

1.123 CMP

CMP		op 1
Description	Compare 2 symbol. It applies a case and space sensitive comparison.	
Related		
Remark		

1.CMP					
Parameter	Type	Values	Comment	Default	Opt
Lpar	symb				
Rpar	symb				
Return					
VAL	NV	-1	if Lpar < of Rpar		
VAL	NV	0	if Lpar == Rpar		
VAL	NV	1	if Lpar >		
RparOnError					
Example					
<pre> CMP (£ciao, "ciao ") >> -1 CMP (£ciao, £) >> 1 CMP (£ciao, NULL) >> 1 CMP (£, £) >> 0 CMP (NULL, £) >> -1 CMP (NULL, NULL) >> 0 CMP (NULL, " ") >> -1 </pre>					



1.124 CHATMSG

CHATMSG		fnc
Description	return the number of outstanding messages in chat	
Related		
Remark		

1.CHATMSG					
Parameter	Type	Values	Comment	Default	Opt
Return					
num	NV		Message num		
OnError					
Example					
!!! Insert a message by CHAT window ; CHATMSG () >>> 1					

1.125 CHATPUT

CHATPUT		fnc
Description	Put a message in chat concatenating symbols	
Related		
Remark		

1.CHATPUT					
Parameter	Type	Values	Comment	Default	Opt
SYMBOL	slist		list of symbol to insert in the current chat line		
Return					
OnError					
Example					



1.126 CHATSHW

CHATSHW		fnc
Description	Return chat message as a text of symb separated by CRLF	
Related		
Remark	Not working in workbench mode	

1.CHATSHW					
Parameter	Type	Values	Comment	Default	Opt
RESET	nv	0;1	1=reset	0	YES
Return					
TEXT	symb		Chat text		
OnError					
Example					
CHATSHW() >> return entire chat text					
CHATSHW(1) >> reset chat text					

1.127 **CHATGET**

CHATGET		fnc
Description	Get the first waiting message from chat	
Related		
Remark		

1.CHATGET					
Parameter	Type	Values	Comment	Default	Opt
Return					
msg	symb				
OnError					
Example					
CHATMSG ()		>>> 2			
CHATGET ()		>>> "Test message insertion in chat window"			
CHATMSG ()		>>> 1			



1.128 CHATCLR

CHATCLR		fnc
Description	Remove all pending message from the chat	
Related		
Remark		

1.CHATCLR					
Parameter	Type	Values	Comment	Default	Opt
Return					
msg	symb				
OnError					
Example					
CHATMSG ()	>>>	2			
CHATCL () ;					
CHATMSG ()	>>>	0			

1.129 **CNS**

CNS		fnc
Description	Show or Hide the console	
Related		
Remark		

1.CNS					
Parameter	Type	Values	Comment	Default	Opt
CMD	symb	£SHOW;£HIDE			
Return					
OnError					
Example					



1.130 STDOUT

STDOUT		fnc
Description	Output text on stdOut	
Related		
Remark		

1.STDOUT					
Parameter	Type	Values	Comment	Default	Opt
TEXT	slist				
Return					
OnError					
Example					
<pre>xml= TREE_SER(tree,NULL,NULL,£TAB); STDOUT(xml) >> in Wiz mode, generate an element in basket STDOUT(£RELOAD) >> in Wiz mode, order the workbench to reload the pka/pki</pre>					

1.131 STDINP

STDINP		fnc
Description	Wait for a text line from stdInp	
Related		
Remark		

1.STDINP					
Parameter	Type	Values	Comment	Default	Opt
Return					
TEXT	symb		Input text		
OnError					
Example					



1.132 YIELD

YIELD		fnc
Description	Release the time slice to system in order to permit the execution of other processes.	
Related		
Remark		

1.133 SLEEP

SLEEP		fnc
Description	Pause the current thread for a period	
Related		
Remark	if period is FLT the function wait an amount of microsecond without explicit release the Cpu	

1.SLEEP					
Parameter	Type	Values	Comment	Default	Opt
period	NV		unsigned =ms; FLT=microsecond		
Return					
OnError					
Example					
SLEEP(500)		>>> wait for half a second			



1.134 WAITCND

WAITCND		fnc
Description	Wait until the exit condition is satisfied	
Related		
Remark		

1.WAITCND					
Parameter	Type	Values	Comment	Default	Opt
SYMBADDRESS	NV		the address\$ of the symbol		
COND	symb	£NULL;£NOTNULL; £EQU;£NOTEQU; £EMPTY;£NOTEMPTY	Condition		
TMO	NV		ms Time out		yes
VALUE	symb		Match value (equ,notEqu)		yes
Return					
RES	NV	1	Condition reached		
OnError					
ERR	NV	-1	Error		
ERR	NV	-2	Time Out		
Example					
<pre>a= 1; t1= WAITCND(\$a, £NOTEQU, 1000, 0); >> -2 if 'a' doesn't become equal to 0 before 1 second >> 1 if 'a' become equal to 0 before 1 second</pre>					

1.135 SMF

SMF		fnc
Description	operation on semaphore	
Related	DOOR	
Remark	There are 255 system semaphore (from 1 to 255) that can be used without creation	

1.SMF		Create – no parameter			
Parameter	Type	Values	Comment	Default	Opt
Return					
PTR to SMF	PTR				
OnError					
Example					
SMF >>> pointer to a new semaphore					



2.SMF		Command on a SMF			
Parameter	Type	Values	Comment	Default	Opt
PTR	PTR		Any Pointer or SMF num (1-255)		
cmd	symb	£LCKS(lock 1 level); £LCKR (unlock 1 level);£RST (fully unlock);£TRY(try to lock);£OWN(return the thread owning); £WAIT(wait until is Locked); £WAITCNT(count of waiting Thr); £LINE(the line of the lock)	Possible command		YES
Return					
Locks	NV		Number of locks for the caller thread		
Status	NV		if cmd N.D		
Thread	NV		£OWN :Thread ID of the owner		
OnError					
Example					

2.SMF	Command on a SMF
<p>With parameters, to make an action on the semaphore. In case the caller asks for SMF lock with £LCKS, if the SMF is locked by someone else, the caller is paused until the semaphore is freed by the legitimate owner. If the SMF is locked by the caller, the level of the block is increased. To fully unlock a semaphore is required the owner to call £LCKR many time the owner have called the lock with £LCKS or to use £RST.</p>	
<pre>SMF () ; >>> SMF PTR</pre>	
<p>Semaphore held by a different caller:</p> <pre>SMF (pntSMF, £LCKS) ; ...wait so the owner unlocks the semaphore... ...after the unlock.... >>> 1</pre>	
<p>Semaphore held by a different caller:</p> <pre>SMF (pntSMF, £TRY) ; >>> 0</pre>	
<p>Semaphore is free:</p> <pre>SMF (pntSMF, £TRY) ; >>> 1 ... SMF (pntSMF, £LCKS) ; >>> 2 (level 2 lock) ... SMF (pntSMF, £LCKR) ; >>> 1 (unlock a level) ... SMF (pntSMF, £LCKS) ; >>> 2 (level 2 lock) ... SMF (pntSMF, £LCKS) ; >>> 3 (level 3 lock) ... SMF (pntSMF, £RST) ; >>> 0 (fully freed)</pre>	
<pre>SMF (smfPtr, £OWN) >> 0 if free, thread id of the locker, if locked SMF (smfPtr, £LINE) >> 0 if free, line where it was locked, if locked</pre>	
<pre>SMF (smfPtr, £WAITCNT) >> return the number of thread awaiting locked semaphore</pre>	



1.136 DOOR

DOOR		fnc
Description	DOOR are synchronization object similar to SMF but can be lock and unLock by different Threads	
Related	SMF	
Remark	There are 255 system DOOR (from 1 to 255)	

1.DOOR		TRY and LCKS			
Parameter	Type	Values	Comment	Default	Opt
DOOR	NV		DOOR number (1-255)		
cmd	symb	£TRY(try to lock); £LCKS(lock 1 level); £LCKR(unlock 1 level);	Possible command		YES
KEY	symb		Key (required for LCKR)		YES
Return					
key	symb		TRY and LOKS return the key		
Locks	NV		Number of locks for the caller thread		
OnError					
RES	symb	ERR			
Example					
<p>With only DOOR number (without cmd) , number of locks are returned</p> <p>With parameters, to make an action on the DOOR. While SMF are operated on thread basis, DOORs work on KEY basis. To Lock and Unlock, also in different threads, you have to use the same key. If the DOOR is locked, new requests with different KEY are queued until the DOOR is unlocked and served on First In basis.</p> <p>TRY and LCKS auto generate a unic key, if the key is null;</p>					

1.137 PWKTASK

PWKTASK		fnc
Description	Execute and manage PWK program as a task	
Related	OSSHELL, OSSTART, OSEXEC	
Remark	The function waits the system to return the control	

1.PWKTASK					
Parameter	Type	Values	Comment	Default	Opt
ACTION	symb	£START;£STOP; £TASKS(ret a TBL of all the running task); £IsTask(test if we are in a TASK or a TSK name); £GETpar(Ret TASK par)			
NAME	symb		The name o the task		YES
COMMANDLINE	symb		The command line		YES
TASK_PAR	symb		parameter for the TASK		YES
Return					
RES	NV		1=ok		
RES(TASKS)	PTR		PTR To TBL with running TASK name		
OnError					
ERR	symb	NULL			
Example					



1.PWKTASK					
Parameter	Type	Values	Comment	Default	Opt
<p>Caller and tasks share their memory, so you can pass a pointer to exchange info. Also the SMF on this pointer are shared.</p> <p>Create a taskTest.pka with an attrib excT and this code in the Main:</p> <pre>excT= PWKTASK(&GETpar); #WHILE(1); TBL_ITM(excT,1,1,TMR); SLEEP(1000); #END;</pre> <p>Create a taskCaller.pka with an attrib excT and this code in the Main:</p> <pre>excT= TBL_NEW(1,1); PWKTASK(&START,&TEST,"tasktest.pka",excT); !!! Launch task passing excT pointer #WHILE(1); tbl1= PWKTASK(&TASKS); !!! Verify if TEST is ON chatput(&tbl1++CRLF++TBL_EXP(tbl1)); TRASH(tbl1); SLEEP(50); chatput(&par..TBL_ITM(excT,1,1)); !!! Write TMR written by taskTest SLEEP(1000); #END;</pre>					

1.138 OSEXEC

OSEXEC		fnc
Description	Pass a command to the operative system executor.	
Related	OSSHELL, OSSTART, PWKEXEC	
Remark	The function waits the system to return the control	

1.OSEXEC					
Parameter	Type	Values	Comment	Default	Opt
command	symb				
Return					
OnError					
Example					



1.139 OSSHELL

OSSHELL		fnc
Description	Execute an OS command.	
Related	OSEXEC, OSSTART, PWKEXEC	
Remark	With £WAIT the program open a doc and waits the closure of the page	

1.OSSHELL					
Parameter	Type	Values	Comment	Default	Opt
action	symb	NULL;£OPEN;£SHOW; £PRINT;£EXPLORE;£WAIT	Action to perform		
cmdPar	symb		Command parameter		
command	slist		command		
Return					
OnError					
Example					
<pre>Print Pippo.odt in PDF via OpenOffice e PDFCreator : OSSHELL(£OPEN,"-pt PDFCreator Pippo.odt","soffice.exe"); Return in attrib t1 the output of a W10 Shell command : t1= OSSTART("wmic process where ""Commandline like '%%PWK-%%'"" get Processid,Caption,Commandline",£GETOUT);</pre>					

1.140 OSSTART

OSSTART		fnc
Description	Start e new process	
Related	OSEXEC, OSSHELL, PWKEXEC	

1.OSSTART					
Parameter	Type	Values	Comment	Default	Opt
cmdLine	symb		Command line		
Par	symb	£STD;£WAIT(wait end); £GETOUT(Wait end and return program stdout)		£STD	YES
Return					
RES	NV				
OnError					
ERR	Symb	NULL			
Example					
to exececute batch command : cmd.exe /c batchFile					
Used to receive text response from a shellcommand					



1.141 CALC

CALC		fnc
Description	Compute a formula	
Related		
Remark		

1.CALC					
Parameter	Type	Values	Comment	Default	Opt
FORMULA	symb				
Return					
RES	symn	the resul			
OnError					
ERR	symb	NULL			
Example					
<pre>a= 2; CALC("1+2*a") >>> 5 a= fpippo; CALC("fphil++(a<<1)") >>> ffilippo</pre>					

1.142 LEN

LEN		fnc
Description	Return the length of a symbol	
Related		
Remark		

1.LEN					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
Len	NV		Number of characters		
OnError					
Example					
LEN("Pippo")	>>>	5			
LEN(£)	>>>	0			
LEN(" ")	>>>	1			
LEN(CRLF)	>>>	2			



1.143 NSP

NSP		fnc
Description	Purge a symbol from initial and final spaces.	
Related		
Remark		

1.NSP					
Parameter	Type	Values	Comment	Default	Opt
PAR	Symb		Input symbol		
Return					
rstl	symb		The symbol purged		
OnError					
Example					
NSP(" Prova Prova ") >>> "Prova Prova"					
NSP(NULL) >>> NULL					

1.144 FST

FST		fnc
Description	Get Or Set the first character of a symbol.	
Related	LST	
Remark		

1.FST					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
NewFst	Symb		If defined set		yes
Return					
RES_GET	symb		A Symb containing only the first char of PAR		
RES_SET	symb		If set the whole symbol		
OnError					
Example					
FST("Prova Prova") >>> "P"					
FST(£Prova,£Ti) >>> £Trova (for new first skip characters other than first)					
FST(£Prova, £) >>> £					



1.145 LST

LST		fnc
Description	Get or Set the last character of a symbol	
Related	FST	
Remark		

1.LST					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
NewLst	symb		If defined Set the last char		YES
Return					
RES_GET	Symb		A symb containing only the last chr of PAR		
RES_Set	symb		If set the whole symbol		
OnError					
Example					
LST("Prova Prova") >>> "a"					
LST(£Prova, £ino) >>> £Provi (for last skip characters other than first)					
LST(£Prova, £) >>> £Prov					

1.146 MID

MID		fnc
Description	Get or SET an internal part of a symbol	
Related		
Remark		

1.MID					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
START	NV		Starting point 1bsd		
LEN	NV		Length in char		YES
SUBSTITUTE	symb		Tne new mart		YES
Return					
RES_GET	symb		The symb extracted		
RES_SET	symb		The whole Symbol		
OnError					
Example					
MID("Prova",3)	>>>	fova			
MID("Prova",3,1)	>>>	fo			
MID("Prova",3,2)	>>>	fov			
MID("Prova",3,3)	>>>	fova			
MID("Prova",2,2)	>>>	fro			



1.147 SPLT

SPLT		fnc
Description	Split a symbol, searching a separator	
Related	TKNZ,TKNZOP,CSV, CSVTBL	
Remark		

1.SPLT		Commento testo			
Parameter	Type	Values	Comment	Default	Opt
symb	symb		Symb to parse		
separator	symb				
command	symb	£LEFT;£LEFTORALL; £RIGHT;£RIGHTORALL		£Left	YES
DIRECTION	symb	£FRW;£REV		£FRW	YES
CASE	NV	0(Insensitive);1(Sensitive)		1	YES
Return					
part	symb		Left or right part		
OnError					
NULL			If separator not found		
Example					
SPLT("Ugo è nato a Roma","è nato a"); >>> "Ugo "					
SPLT("Ugo è nato a Roma","è nato a",£LEFT); >>> "Ugo "					
SPLT("Ugo è nato a Roma","è nato a",£RIGHT); >>> " Roma"					

1.148 TKNZ

TKNZ		fnc
Description	Tokenize a symbol using a list of separators	
Related	TKNZOP,SPLT,CSV, CSVTBL	
Remark		

1.TKNZ					
Parameter	Type	Values	Comment	Default	Opt
PAR	Symb		Input symbol		
Separators	Slist		List of separators		
Return					
List	PTR		Pointer to a LIS of tokenized symbol		
OnError					
Example					
TKNZ ("A + B = C", "+");		>>>	List ("A ", " B = C")		
TKNZ ("A + B = C", "+", "=");		>>>	List ("A ", " B ", " C")		



1.149 TKNZOP

TKNZOP		fnc
Description	Tokenize a symbol using a list of separator. Maintain the separator as element of the list.	
Related	TKNZ,SPLT,CSV,	
Remark		

1.TKNZOP					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		Input symbol		
Separators	slist		List of separators		
Return					
List	PTR		Pointer to a LIS of tokenized symbol and separators		
OnError					
Example					
TKNZ ("A + B = C","+") ; >>> List ("A ","+", " B = C")					
TKNZ ("A + B = C","+", "=") ; >>> List ("A ","+", " B ","=", " C")					

1.150 CSV

CSV		fnc
Description	Broke a symbol using the default or the provided separator	
Related	TKNZ,TKNZOP,SPLT,CSVTL	
Remark		

1.CSV					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		Input symbol		
Separator	symb		Separator	,	yes
Return					
List	PTR		Pointer to a LIS		
OnError					
ERR	NULL				
Example					
<pre> CSV("A,B,C"); >>> List ("A","B","C") CSV("A,B;C",";"); >>> List ("A,B","C") a= "A"++CRLF++"B"++CRLF++"C" CSV(a) >>> List ("A","B","C") </pre>					



1.151 CSVTBL

CSVTBL		fnc
Description	Broke a symbol using separator for column and for rows preserving values content in block	
Related	TKNZ,TKNZOP,SPLT, CSV	
Remark	Return a TBL	

1.CSVTBL					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		Input symbol		
COL_Separator	symb		Column separator	,	YES
ROW_Separator	symb		Rows separator	<CRLF>	YES
BLOCK_indicator	simb		Block indicatore		YES
Return					
TBL	PTR		Pointer to a TBL		
OnError					
ERR	NULL				
Example					
CSVTBL("A;B;C d;e;f",";"," "); >>> TBL("A","B","C") ("d","e","f") To escape BLOCK_indicator double it					
CSVTBL("`a;b`;b;c d;`e;f`;f",";"," ","`") >>> TBL("a;b","b","c") ("d","e;f","f")					

1.152 SRCH

SRCH		fnc
Description	Search a symbol in an other.	
Related		
Remark	Case sensitive	

1.SRCH					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
search	symb		Symb to search		
Start	NV		Starting point 1bsd		YES
CASE	NV	0 (case insensitive); 1 (case sensitive)		1	YES
REVERSE	NV	£FRW(forward search); £RVS (reverse search);	0=FRW, 1=RVS	0	YES
Return					
POS	NV		Position 1bsd		
POS	NV	0	Not found		
OnError					
Example					
<pre>SRCH("ABC", "A"); >>> 1 t1=SRCH("CAROLINA", "A"); >>> t1=2 t1=SRCH("CAROLINA", "A", t1+1); >>> t1=8 SRCH("CAROLINA", "a") >>> t1=0 (default case sensitive)</pre>					



1.153 CAT

CAT		fnc
Description	Concatenate a list of symbols	
Related		
Remark		

1.CAT					
Parameter	Type	Values	Comment	Default	Opt
PAR	slist		symbol to concatenate		
Return					
RES	symb		concatenation of symbol		
OnError					
Example					
<pre>a=CAT("Pippo ", "ha ", 5, " anni!"); >>> a="Pippo ha 5 anni!" equivalente a : "Pippo "++"ha "++5++" anni!"</pre>					

1.154 MTCH

MTCH		fnc
Description	Search a symbol in an other with similar match too.	
Related	CMP	
Remark	The comparison is totally case insensitive and space insensitive.	

1.MTCH					
Parameter	Type	Values	Comment	Default	Opt
PAR1	symb				
PAR2	symb				
Return					
RES	symb	£EXACT	if two symbol are identical		
RES	symb	£CONTAINS	PAR1 contains PAR2		
RES	NV		If no match, the system search if the character of a symbol are present in the other, and return a NV with the thousandths of similarity.		
OnError					
Example					
MTCH("Torta di mele","Tortadimele"); >>> £EXACT					
MTCH("Torta di mele","di MELE"); >>> £CONTAINS					
MTCH("Pippo","POIOPPO"); >>> 1000					

**1.155 LWR**

LWR		fnc
Description	Convert all Upper characters of the symbol to lower	
Related	UPR	
Remark		

1.LWR					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		Symbol to "LOWERIZE"		
Return					
rsIt	symb		Lowerized symbol		
OnError					
Example					
LWR ("APPlE") >>> "apple"					

1.156 UPR

UPR		fnc
Description	Convert all Lower characters of the symbol to upper	
Related	UPR	
Remark		

1.UPR					
Parameter	Type	Values	Comment	Default	Opt
PAR	symp		Symbol to "UPPERIZE"		
Return					
rslt	symp		Upperized symbol		
OnError					
Example					
LWR ("APPlE") >>> "APPLE"					

**1.157 RTF2TXT**

RTF2TXT		fnc
Description	Convert RTF to standard Text	
Related		
Remark		

1.RTF2TXT					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb				
Return					
rsIt	symb		TEXT		
OnError					
Example					

1.158 PKGPTH

PKGPTH		cst
Description	Return the Package path or NULL	
Related		
Remark		

1.PKGPTH					
Parameter	Type	Values	Comment	Default	Opt
Return					
rslt	symb		TEXT		
OnError					
Example					
PKGPTH	>> "C:\PWK-TMP\PWK-ISP-07-1589814237-8276\" temp package unzip dir				



1.159 PWKPTH

PWKPTH		cst
Description	Return the path to POWER-KI directory	
Related		
Remark		

1.PWKPTH					
Parameter	Type	Values	Comment	Default	Opt
Return					
rslt	symb		TEXT		
OnError					
Example					
PWKPTH	>> "C:\Program Files (x86)\XPLAB\POWER-KI\" executable directory				

1.160 EXECMODE

EXECMODE		cst
Description	Return 1 if in exec mode , otherwise (WBK) return 0	
Related		
Remark		



1.161 ALIASPTH

ALIASPTH		fnc
Description	Set or Get an alias for a ROOT item valid only inside the current thread	
Related		
Remark	With only one parameter return how PWK apply the conversion to the symb	

1.ALIASPTH					
Parameter	Type	Values	Comment	Default	Opt
ALIAS	symb				
TOITEM	symb		if defined but NULL, delete the alias, elsewhere define. If N.D. return the alias		yes
Return					
rsit	symb		the alias (could be NULL if reset)		
OnError					
ERR	symb	NULL			
Example					
<pre>ALIASPTH("\MP","\MP1") !! set \MP as alias of \MP1 ; ALIASPTH("\MP","\MP2") !! change and set \MP as alias of \MP2 ; r=ALIASPTH("\MP\PAG") >>> r="\MP2\PAG" r=ALIASPTH("id@\MP\PAG") >>> r="id@\MP2\PAG" ALIASPTH("\MP"); !! Delete the alias Remember ONLY ROOT ITEM can be aliased</pre>					

1.162 TRIG

TRIG		fnc
Description	Define a trigger, linked to an EXO or MTDH.	
Related	TRIGPAR(deprecated), TRIGSET (deprecated)	
Remark	if EXO TRIGPAR names are different use the = to establish the relation)	

1.TRIG					
Parameter	Type	Values	Comment	Default	Opt
NAME	symb		EXO or MTDH to execute		
TRIGPAR	enum		Trigger pars		YES
PAR	symb		A symb to pass as _PAR_		YES
Return					
RES	PTR		Pointer to the trig		
OnError					
Example					
<p>TRIG are the way to invoke EXO from asynchronous event like SOK connection, WEB .. Depending from the cause of its activation the trig can pass to the exo different parameters the have standard name.</p> <p>You can find these names in the documentation of function that emit the trig.</p> <p>If the called EXO make use of any of this parameter you should declare it and , if the EXO use a different name the equivalent name.</p> <p>You can also set a default parameter to be passed to the EXO, with _PAR_ key word.</p> <pre>TRIG(£\MyTrig, "TRG_XITM, TRG_KB1=KB1", OPAQUE);</pre> <p>note: if OPAQUE is defined and _PAR_ is not in the TRIGPAR list is added by default by PWK. OPAQUE is passed to the trig as _PAR_.</p>					



1.163 TRIGSET

TRIGSET		fnc
Description	(deprecated) Define the relationship between trigger parameters and real variable of the linked function.	
Related	TRIG, TRIGPAR	
Remark	The TRIGpar code name depend of the function that use the TRIG	

1.TRIGSET					
Parameter	Type	Values	Comment	Default	Opt
TRIG	PTR		Pointer to TRIG		
TRIGparName	symb		TRIG par code name		
PAR	symb		EXO or MTDH par		
Return					
OnError					
Example					

1.164 TRIGPAR

TRIGPAR		fnc
Description	(deprecated) Set the trigger pars (comma separated enum) and define the relationship with the EXO pars	
Related	TRIG, TRIGSET(deprecated)	
Remark	if EXO params names are different use the = to establish the reation	

1.TRIGPAR					
Parameter	Type	Values	Comment	Default	Opt
TRIG	PTR		Pointer to TRIG		
TRIGPAR	enum		Trigger pars		
Return					
OnError					
Example					
TRIGPAR(trg, "TRG_XITM, TRG_KB1=KB1);					



1.165 LIC

LIC		fnc
Description	Get information from the License file	
Related		
Remark		

1.LIC					
Parameter	Type	Values	Comment	Default	Opt
SELECTOR	symb	£LVL(level); £DEMO(Is demo?); £TS(Time Stamp); £KEYID(Required Id)			
Return					
INFO	symb				
OnError					
Example					

1.166 THRINF

THRINF		fnc
Description	Return information about thread	
Related		
Remark		

1.THRINF					
Parameter	Type	Values	Comment	Default	Opt
WHAT	symb	£ID (thread id);£NAM(thread Name);£LINE (the idx of the line in esecution); £CALLID(the caller thread ID);£INFO(get/set a symbol); £TBL(a table with ID,NAM,CALLID,INFO,WSown,SYserr of all treads); £SYserr;£wsOwn(ID of the thread block this)			YES
ID	NV		if N.D. the current thread is intended		YES
Value	symb		set Value for info		YES
Return					
Info	symb		the required information		
OnError					
Example					
<pre>callId >> ID of the thread lauch this sysErr >> number of system errors occurred during thread execution Wsown >> ID of the thread block this (usually locking a SMF, requested by this too) THRINF(£INFO,thrId,25) >> save in the thread internal value 25 THRINF(£INFO,thrId) >> 25</pre>					



1.167 THRSYM

THRSYM		fnc
Description	Provides a Thread Local storage for Symbol	
Related		
Remark		

1.THRSYM					
Parameter	Type	Values	Comment	Default	Opt
SYMB	symb		The symb name		
VALUE	symb		set value		YES
SPACE	symb	£LOC;£THREAD		THREAD	YES
Return					
RES	symb		The value		
OnError					
Example					
<pre>THRSYM(£var1, "test") >> create an internal attrib 'var1' in current thread with value "test" °var1°="test" >> comando equivalente THRSYM(£var1) >> "test" : read value of the internal var1 °var1° >> "test" : comando equivalente >> var1 is visible in all the MTHD called by EXEC by the THREAD THRSYM(£var1, "test", £LOC) >> create an internal attrib 'var1' in current thread with value "test" °var1°="test" >> comando equivalente THRSYM(£var1, NULL, £LOC) >> "test" : read value of the internal var1 °var1 >> "test" : comando equivalente >> var1 isn't visible in all the MTHD called by EXEC by the THREAD</pre>					

1.168 THRPRI

THRPRI		fnc
Description	Set Thread execution priority	
Related	THRINF	
Remark		

1.THRPRI					
Parameter	Type	Values	Comment	Default	Opt
PRI	symb	£LOW;£NRM;£FST(fast); £HGH(hight); £RTM(realTime)	Priority		
THR_ID	symb		thread id	current	YES
Return					
RES	NV		The priority		
OnError					
RES					
Example					



1.169 PTRTYP

PTRTYP		fnc
Description	Return the TYP of a PTR if WITHLIB par is true in the form of TYP_LIB	
Related		
Remark		

1.PTRTYP					
Parameter	Type	Values	Comment	Default	Opt
POINTER	PTR				
WITHLIB	symb	TRUE			YES
Return					
TYP(_LIB)	symb		Pointer TYP, if WITHLIB TYP_LIB(connected by underscore)		
OnError					
ERR	symb	NULL			
Example					
<pre>PTR=TBL_NEW(1,1); PTRTYP(PTR) >> "TBL" PTR=LIS_NEW(); PTRTYP(PTR) >> "LIS"</pre>					

1.170 PTRDUP

PTRDUP		fnc
Description	Increment the pointer reference count	
Related	TRASH	
Remark	the pointer should be trashed one time more for each DUP but it is safe against deletions	

1.PTRDUP					
Parameter	Type	Values	Comment	Default	Opt
POINTER	PTR				
Return					
POINTER	PTR				
OnError					
ERR	symb	NULL			
Example					
<p>Thread 1 creates PTR Thread 2 trash PTR PTR is thrashed</p> <p>Thread 1 creates PTR Thread 2 use PTRDUP(PTR) Thread 1 TRASH(PTR) PTR still exists Thread 2 TRASH(PTR) PTR is thrashed.</p> <p>Thread 1 creates PTR Thread 2 use PTRDUP(PTR) Thread 2 TRASH(PTR) PTR is thrashed</p>					



1.171 PTRLIS

PTRLIS		fnc
Description	Return the LIS of existing PTR	
Related		
Remark	The auxiliary element of each item contains the THREAD id, separated by "-" from the line number where the pointer was created, separated by "-"	

1.PTRLIS					
Parameter	Type	Values	Comment	Default	Opt
Return					
PTR	symb		PTR to the Lis (- line number)		
OnError					
Example					
ptr=PTRLIS();					
LIS_POS(ptr,1) >> the oldest pointer allocated in the program					
LIS_POS(ptr,£ATT) >> <id of the thread allocated it>-<code line where allocates>					

1.172 THIS

THIS		fnc
Description	Return the address (\$) of the current code item of the main TREE	
Related		
Remark		

1.THIS					
Parameter	Type	Values	Comment	Default	Opt
Return					
PTR	NV				
OnError					
Example					
<pre>t=TREE_OPN(); TREE_ITM(t,THIS,£VAL) >> text of code where you write this</pre>					



2

SYMB

2.1 SYMB_NRM

SYMB_NRM		fnc
Description	Normalize a symbol. Upper case and remove all char having value les or equal to spaces .	
Related		
Remark		

1.SYMB_NRM					
Parameter	Type	Values	Comment	Default	Opt
PAR	symp		Symb to normalize		
ASSPACE	symp		The first char is used as replacement for SPACE		YES
DISCARD	symp		The contained char in found in par are discarded		YES
Return					
RES	symp		Normalized symbol		
OnError					
Example					
<pre>a = " Prova bella "; SYMB_NRM(a); >>> "PROVABELLA"</pre>					



2.2 SYMB_INDXX

SYMB_INDXX		fnc
Description	Index of a symbol inside a comma separated list contained in a symbol	
Related		
Remark		

1.SYMB_INDXX					
Parameter	Type	Values	Comment	Default	Opt
SEARCH	symb		Symbol to find (single word without spaces inside)		
IN	enum		enumeration of comma separated values		
SEP	NNV		Separator (one CHR)	,	YES
Return					
POS	NV		1bsd position		
POS	NV	0	Not Found		
OnError					
Example					
<pre>SYMB_INDXX("tre","uno,due tre,quattro"); >>> 2 SYMB_INDXX("tr","uno,due tre,quattro"); >>> 0 (not found)</pre>					

2.3 SYMB_SLCT

SYMB_SLCT		fnc
Description	Giving a NV as index (1bsd) return the symbol in the slist or NULL	
Related		
Remark		

1.SYMB_SLCT					
Parameter	Type	Values	Comment	Default	Opt
INDEX	NV		1bsd		
PARLIST	slist		list of symb		
Return					
RES	symb		symb at the position		
RES	NV	NULL	not found		
OnError					
Example					
SYMB_SLCT(3,"uno","due","tre"); >>> "tre"					



2.4 SYMB_DCD

SYMB_DCD		fnc
Description	Decode a symbol contained in a enumeration with the counterpart at the same position in a list.	
Related		
Remark	case insensitive comparison is applied	

1.SYMB_DCD					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		Symbol to decode (a single word without spaces inside)		
PARENUM	enum		Symbol containing a comma separated list of symbols		
PARLIST	slist		List of symbol		
Return					
RES	symb		the found symb in PARLIST		
RES	NV	NULL	Not found		
OnError					
Example					
SYMB_DCD(&uno, "uno,due,tre",&primo,&secondo,&terzo) >>> "primo"					

2.5 SYMB_RPLC

SYMB_RPLC		fnc
Description	Replace (or remove) a symbol part	
Related		
Remark	case insensitive comparison is applied	

1.SYMB_RPLC					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		Symbol to Process		
SEARCHFOR	symb		Symbol to replace		
REPLACEWITH	symb		Replacement or NULL		
TIMES	NV		How many times	NULL	YES
STARTPOINT	NV		Starting from char (1bsd)		YES
CASE	NV	0(Insensitive);1(Sensitive)		1	YES
Return					
RES	symb		the symb with replacement applied		
RES	NV	NULL	Not found		
OnError					
Example					
SYMB_RPLC ("?abc?d", "?", "1") >>> "1abc1d"					



2.6 SYMB_BLKSEQ

SYMB_BLKSEQ		fnc
Description	Retrive inside a symbol the sequence of symbol not separated or separated by space provided as a slis	
Related	SYMB_BLK	
Remark		

1.SYMB_BLKSEQ					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		Symb in witch search		
CASEtype	NV	0 (insensitive);1 (sensitive);	Type of comparison		
Sequence	slis		Sequence of symbol to search		
Return					
RES	PTR		Pointer to a symb BLK		
OnError					
RES	symb	NULL			
Example					
<pre>a = " testo prima < br \> testo dopo "; blk=SYMB_BLKSEQ(a,0,"<","br","\\",">"); SYMB_BLK(blk,fBEF) = " testo prima " SYMB_BLK(blk,fAFT) = " testo dopo " SYMB_BLK(blk,fVAL) = "< br \>" b = " testo prima <br\> testo dopo "; blk=SYMB_BLKSEQ(b,0,"<","br","\\",">"); SYMB_BLK(blk,fBEF) = " testo prima " SYMB_BLK(blk,fAFT) = " testo dopo " SYMB_BLK(blk,fVAL) = "<br\>"</pre>					

2.7 SYMB_BLKBLK

SYMB_BLKBLK		fnc
Description	giving a starting and ending LIS of symbol retrieve inside a symb, the contained BLK	
Related	SYMB_BLK	
Remark		

1.SYMB_BLKBLK					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		Symb in witch search		
CASEtype	NV	0(insensitive); 1(sensitive);	Type of comparison		
StartSequence	PTR		to LIS of start Sequence		
EndSequence	PTR		to LIS of end Sequence		
Return					
RES	PTR		Pointer to a symb BLK		
OnError					
RES	symb	NULL			
Example					
<pre> a = " testo prima < testo contenuto > testo dopo "; blk=SYMB_BLKBLK(a,0,LIS_NEW("<"),LIS_NEW(">")); SYMB_BLK(blk,&BEF) = " testo prima " SYMB_BLK(blk,&AFT) = " testo dopo " SYMB_BLK(blk,&VAL) = "< testo contenuto >" SYMB_BLK(blk,&CNT) = " testo contenuto " </pre>					



2.8 SYMB_BLK

SYMB_BLK		fnc
Description	Get the element of a symb BLK	
Related	SYMB_BLKSEQ	
Remark		

1.SYMB_BLK					
Parameter	Type	Values	Comment	Default	Opt
BLK	PTR		Pointer to a symb BLK		
SELECTOR	symb	£BEF;£AFT;£VAL; £CNT(content); £BEG(Begin position 1bsd); £END(Begin position 1bsd);	if N.D. £CNT is assumed		yes
Return					
RES	symb		Pointer to a symb BLK		
OnError					
RES	symb	NULL			
Example					
<pre>a = " testo prima < br \> testo dopo "; blk=SYMB_BLKSEQ(a,0,"<","br","\\",">"); SYMB_BLK(blk,£BEF) = " testo prima " SYMB_BLK(blk,£AFT) = " testo dopo " SYMB_BLK(blk,£VAL) = "< br \>" b = " testo prima <br\> testo dopo "; blk=SYMB_BLKSEQ(b,0,"<","br","\\",">"); SYMB_BLK(blk,£BEF) = " testo prima " SYMB_BLK(blk,£AFT) = " testo dopo " SYMB_BLK(blk,£VAL) = "<br\>"</pre>					

3 DB



3.1 DB_OPN

DB_OPN		fnc
Description	Open a Data Base connection	
Related		
Remark	delete with TRASH	

1.DB_OPN					
Parameter	Type	Values	Comment	Default	Opt
TYPE	symb	£MYSQL;£ODBC; £SQLITE			
IP	symb		IP address or host name		
USER	symb		User name		YES
PASSWORD	symb		User Password		YES
SCHEMA	symb		Default schema name		YES
Return					
RES	PTR		a pointer to the db		
OnError					
Example					
DB_OPN (£MYSQL, "127.0.0.1", £root, £root)			>>> ptr to mySql communication		
DB_OPN (£sqlite, "local.db")			>>> ptr to sqLite on file local.db		

3.2 DB_QRY

DB_QRY		fnc
Description	Make a query to DB	
Related	DB_GET	
Remark	If a result is returned it must be TRASH after use	

1.DB_QRY					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		a valid PTR to DB	PAR	PTR
PARLIST	slist		a slist constituting the query	PARLIST	slist
Return					
RES	PTR		a pointer to query result to use with DB_GET		
OnError					
RES	symb	NULL	Connection error	RES	symb
Example					
dbRes=db_qry(db,"SELECT count INTO table");					



3.3 DB_GET

DB_GET		fnc
Description	Get data from a query result	
Related	DB_QRY	
Remark		

1.DB_GET		Rows and Columns num			
Parameter	Type	Values	Comment	Default	Opt
DBQRYRES	PTR		a PTR to DB query result		
ACTION	symb	£ROW(rows num); £COL(columns num);£RSLT(result)			
Return					
RES	NV		Row or column num . If £RSLT RES indicate the affected rows if >=0 or an error if negative		
OnError					
Example					
lRes= DB_QRY(dbPtr."SELECT * FROM client"); !! TBL 20 columns, 5 rows					
DB_GET(lRes,£ROW)		>>> 5 : number of result rows			
DB_GET(lRes,£COL)		>>> 20 : number of result columns			
DB_GET(lRes,£RSLT)		>>> 5 : affected rows (the listed rows in this case)			

2.DB_GET		Field content			
Parameter	Type	Values	Comment	Default	Opt
DBQRYRES	PTR		a PTR to DB query result		
ACTION	symb	£FLD			
COL	symb		the num 1bsd or the name of the field		
ROW	NV		the num 1bsd of the row		
Return					
RES	symb		the content of the field		
OnError					
ERR	symb	NULL			
Example					
<pre>lRes= DB_QRY(dbPtr."SELECT * FROM client"); !! TBL 20 columns, 5 rows DB_GET(lRes,£FLD,£name,1) >>> "luca"</pre>					

3.DB_GET		A TBL containing the results			
Parameter	Type	Values	Comment	Default	Opt
DBQRYRES	PTR		a PTR to DB query result		
ACTION	symb	£TBL			
Return					
RES	symb		PTR to the LIS		
OnError					
ERR	symb	NULL			
Example					

4.DB_GET		Row Lis			
Parameter	Type	Values	Comment	Default	Opt
DBQRYRES	PTR		a PTR to DB query result		
ACTION	symb	£ROWLIS			
ROW	NV		the num 1bsd of the row		
Return					
RES	PTR		PTR to the LIS		
OnError					
ERR	symb	NULL			
Example					
<pre>lRes= DB_QRY(dbPtr."SELECT * FROM client"); !! TBL 20 columns, 5 rows DB_GET(lRes,£ROWLIS,1) >>> List(1,"luca","Zampolon",28,5,1976,"Fresia"...)</pre>					

5.DB_GET		Columns Lis			
Parameter	Type	Values	Comment	Default	Opt
DBQRYRES	PTR		a PTR to DB query result		
ACTION	symb	£COLLIS			
COL	symb		the num 1bsd or the name of the column		
Return					
RES	symb		PTR to the LIS		
OnError					
ERR	symb	NULL			
Example					
<pre>lRes= DB_QRY(dbPtr."SELECT * FROM client"); !! TBL 20 columns, 5 rows DB_GET(lRes,£COLLIS,£name) >>> List("luca","pino","lino","arturo","tristano")</pre>					



6.DB_GET		Column name			
Parameter	Type	Values	Comment	Default	Opt
DBQRYRES	PTR		a PTR to DB query result		
ACTION	symb	£NAM			
COL	NV		the num 1bsd field		
Return					
RES	symb		the name of the column		
OnError					
Example					
<pre>lRes= DB_QRY(dbPtr."SELECT * FROM client"); !! TBL 20 columns, 5 rows</pre>					
<pre>DB_GET(lRes,£NAM,1) >>> £ID</pre>					
<pre>DB_GET(lRes,£NAM,2) >>> £name</pre>					

4 TREE



4.1 TREE_OPN

TREE_OPN		fnc
Description	Create or open a tree	
Related		
Remark	if no PAR is provided then the program tree is opened.	

1.TREE_OPN					
Parameter	Type	Values	Comment	Default	Opt
PAR	symb		£NEW for an empty tree or a path to an existing file containing a tree definition.		YES
STRICT	symb	0(NO);1(yes)	Strict XML flag	0	YES
Return					
RES	PTR		Pointer to TREE		
OnError					
Example					
TREE_OPN() >>> actual program tree pointer. Pointer has to have destroyed when you finished					
TREE_OPN (£NEW) >>> poiter to new independant tree					
TREE_OPN ("c:\PWK-PRG\test.pka") >>> pointer to pka					

4.2 TREE_PARSE

TREE_PARSE		fnc
Description	Parse a unicode text	
Related		
Remark		

1.TREE_PARSE					
Parameter	Type	Values	Comment	Default	Opt
TREE	symb		A pointer to tree		
TEXT	symb		The text to parse		
STRICT	symb	0(NO);1(yes) ;2(HTML)	(0/1)=Strict XML flag(0/1). (2)=HTML	0	YES
Return					
RES	NV	1	Ok		
OnError					
RES	NV	ERR			
Example					
<pre>txt("<xml....>" >> text XML description of an item TREE_PARSE(tree,txt) >> add item, described by txt, in tree</pre>					



4.3 TREE_SER

TREE_SER		fnc
Description	Serialize a tree or an item of a tree	
Related	TREE_PARSE	
Remark		

1.TREE_SER					
Parameter	Type	Values	Comment	Default	Opt
TREE	symb		A pointer to tree		
ITEM	symb		if N.D. all the tree is serialized		YES
FILE	symb		if N.D. the result return the serialization		YES
FLAG	enum	£ENC(encode entity); £TAB(output with tab); £HTML(html output); £KRP(crypto)	enc=encode entity; tab= output with tab; html=html output, krp=crypto	NULL	YES
Return					
RES	symb	NULL	serialized on file		
RES	symb		I serialized in memory		
OnError					
RES	symb	ERR			
Example					
<pre>TREE_SER(tree) >> "<XML...>" : it returns a text with the serialization in xml of all the program TREE_SER(tree,itemPtr) >> "<XML...>" : it returns a text with the serialization in xml of all the program</pre>					

4.4 TREE_SAV

TREE_SAV		fnc
Description	Save a tree in a file	
Related		
Remark		

1.TREE_SAV					
Parameter	Type	Values	Comment	Default	Opt
TREE	PTR		A pointer to a tree		
FILE	symb		the file name		
FLAG	enum	TAB(output with tab); £HTML(html output); £KRP(crypto)	tab= output with tab; html=html output, krp=crypto	NULL	YES
Return					
RES	PTR		Pointer to TREE		
OnError					
RES	symb	NULL			
Example					
TREE_SAV(tree,"test.pka") >> save tree in file test.pka					



4.5 TREE_PTH

TREE_PTH		fnc
Description	Retrieve a TREE_ITM address giving the tree path	
Related		
Remark	if PATH is NULL or \ the root item is returned	

1.TREE_PTH					
Parameter	Type	Values	Comment	Default	Opt
TREE	PTR		A pointer to a tree		
PATH	symb		the tree path		
StartAddress	NV		item address from witch start		YES
Return					
RES	NV		Item address		
OnError					
RES	symb	NULL			
Example					
<pre>tree=TREE_OPN(); >> open the tree of the current program TREE_PTH(tree,"") >> pointer to the PWK item in the program TREE_PTH(tree,"\Main") >> pointer to the \Main item in the program</pre>					

4.6 TREE_ITM

TREE_ITM		fnc
Description	Tree Item manipulation	
Related		
Remark		

1.TREE_ITM					
Parameter	Type	Values	Comment	Default	Opt
TREE	PTR		A pointer to a tree		
ITM	symb		The Item name or address or £NEW or £ NEWINROOT		
WHAT	symb	£VAL(Value);£PRV(previous); £NXT(next);£UPR(The Upper Item); £DUP(duplicate); £DEL(delete); £PTH(return the full item Path); £ITM(ItemName); £PFX(prefix);£LBL(label); £TYP(type);£XIT(xtype); £SUB(subLis);£ATT(attLis); £ATY(Attribute Type); £IDX(the Index of ITM); £PTR(pointer of item); £ADD(add \$ITM in val at sub); £ADDB(add \$ITM in val BEFORE); £ADDA(add \$ITM in val AFTER)			
VAL	symb		if ITM=£NEW or WHAT=£XIT(set) then VAL should be one of: £ELEM, £ATTR, £TEXT, £XCMD, £CDTA, £DTDE, £CMNT		yes
AttVal	Symb				YES
Return					
RES	symb		Result according to parameter		
OnError					
RES	symb	NULL			
Example					

**1.TREE_ITM**

```
tree=TREE_OPN();           >> pointer to current program
itm=TREE_PTH(tree,"");      >> pointer to first item in tree (\pwk)
itm=TREE_PTH(tree,"\pwk");  >> equivalent to previous code line
```

In usual pka, pwk contains items (VAR) Editor and Executor

```
sItm= TREE_ITM(tree,itm,£sub) >> pointer to first item in pwk : Editor
sItm= TREE_ITM(tree,sItm,£nxt) >> p. to next object in the space of Editor : Executor
TREE_ITM(tree,sItm,£TYP)      >> VAR : the type of Executor
```

If the attributes are missing from an item and you want to add them:

```
attP=TREE_ITM(tree,£NEW,£XIT,£ATTR);
TREE_ITM(tree,attP,£ITM,"varDbg");
TREE_ITM(tree,attP,£LBL,"varDbg");
TREE_ITM(tree,attP,£VAL,"100");
```

```
TREE_ITM(tree,dstItm,£ATT,attP);
```

5 DOC



5.1 DOC_DOC

DOC_DOC		fnc
Description	Create a Doc PTR and open it OR open a doc in a window OR print	
Related		
Remark	With £WAIT the document is open in a separate window. The application stops until this windows is open.	

1.DOC_DOC		Create a pointer to doc			
Parameter	Type	Values	Comment	Default	Opt
TYPE	symb	£OO(OpenOffice); £DEF(Default)		£DEF	
DOCname	symb		Document name		YES
ACTION	symb	£SHOW(Open in a window); £PRINT; £WAIT(Open and wait for closure); £CREATE	With £CREATE (or N.D.)and DOCname defined, the doc is opened.	£CREATE	YES
Return					
DOCptr	PTR		If DOCname is Null or ACTION is NULL		
RES	symb	NULL	if £SHOW,£WAIT,£PRINT ACTION		
RES	symb	NULL	If DOC could not be open		
OnError					
RES	symb	NULL			
Example					
docPtr=DOC_DOC (£00, "text.odt") >> open "text.odt" for read or modification					

5.2 DOC_TPL

DOC_TPL		fnc
Description	Create a document from a template	
Related		
Remark	The name of destination document (without a path the document is stored in \windows\temp)	

1.DOC_TPL					
Parameter	Type	Values	Comment	Default	Opt
SourceDoc	symb		The name of the source document		
DestDoc	symb		The name of destination document (without a path the document is stored in \ windows\temp)		
TemplateDir	symb		The template dir		YES
Return					
CreateDoc	symb		The name of the template		
OnError					
Example					
DOC_TPL("template.odt", "c:\report\rep01.odt", PKGPTH) >> "c:\report\rep01.odt"					
This command generates file "c:\report\rep01.odt" as copy of the template file "template.odt" contained in the package.					



5.3 DOC_OPN

DOC_OPN		fnc
Description	Open a DOC	
Related	DOC_DOC	
Remark	Without the DOCNAME return the open status. If DOCNAME defined the old document is closed and the new one is opened.	

1.DOC_OPN					
Parameter	Type	Values	Comment	Default	Opt
DOCptr	PTR		Ptr to doc		
DOCNAME	symb		New doc name		yes
Return					
RES	NV	1	Doc is open		
RES	NV	0	Doc not open		
OnError					
RES	symb	NULL			
Example					

5.4 DOC_CLS

DOC_CLS		fnc
Description	Close a DOC	
Related	DOC_SAV	
Remark		

1.DOC_CLS					
Parameter	Type	Values	Comment	Default	Opt
DOCptr	PTR		PTR to doc		
Return					
OnError					
Example					



5.5 DOC_TBL

DOC_TBL		fnc
Description	Retrieve a table inside DOC	
Related		
Remark		

1.DOC_TBL					
Parameter	Type	Values	Comment	Default	Opt
DOCptr	PTR		PTR to DOC		
TABLE	symb		Table Name		
Return					
TABLEptr	PTR		PTR to table		
OnError					
RES	NULL				
Example					
tblPtr=DOC_TBL(docPtr,"Table1") >> docPtr is a pointer to a ODT text document, with a table named "Table1". You can name a table, entering in the property or simply write the name in the first cell of the table (high left), The same command creates also the pointer to a sheet of a ODS datasheet.					

5.6 DOC_TBLDUP

DOC_TBLDUP		fnc
Description	Duplicate a table and insert before or after a table	
Related		
Remark	if n.d. the REF_TABLE the duplicated table is used as reference for insertion	

1.DOC_TBLDUP					
Parameter	Type	Values	Comment	Default	Opt
TABLEptr	PTR		PTR to Table to duplicare		
NAME	symb		Table Name		YES
LINES	NV		Nuber of line between	0	YES
DIRECTION	symb	£AFTER;£BEFORE		£AFTER	YES
REF_TABLEptr	PTR		PTR to TABLE for insertion		YES
Return					
TABLEptr	PTR		PTR of duplicated table		
OnError					
RES	NULL				
Example					
<pre> !!! dstDos is a valid doc file, with a t_init and a t_a tables ; initDT=DOC_TBL(dstDoc,"t_init"); aDT=DOC_TBL(dstDoc,"t_a"); !!! Insert a new t_a table after t_init ; DOC_TBLDUP(aDT,"t_a_1",2,£AFTER,aDT); DOC_SAV(dstDoc); DOC_CLS(dstDoc); </pre>					



5.7 DOC_TBRLMV

DOC_TBRLMV		fnc
Description	Remove a Table from the doc and also TRASH the PTR	
Related	DOC_TBL, DOC_TBLDUP	
Remark	i	

1.DOC_TBRLMV					
Parameter	Type	Values	Comment	Default	Opt
TABLEptr	PTR		PTR to Table to duplicare		
Return					
RES	NV	1	Done		
OnError					
RES	NULL		Error		
Example					
<pre>!!! dstDos is a valid doc file, with a t_a table ; aDT=DOC_TBL(dstDoc,"t_a"); DOC_TBRLMV(aDT); DOC_SAV(dstDoc); DOC_CLS(dstDoc);</pre>					

5.8 DOC_UF

DOC_UF		fnc
Description	Set or Get User field	
Related		
Remark		

1.DOC_UF					
Parameter	Type	Values	Comment	Default	Opt
DOCptr	PTR		Pointer to DOC		
UFName	symb		User Field Name		
VALUE	symb		User Filed Name		YES
Return					
VALUE	symb		User Field Value if get		
OnError					
Example					
docPtr is pointer to a text ODT document that contains a user field "clientId" DOC_UF(docPtr,&clientId,"Gerolamo") >> set value of user field "clientId" to "Gerolamo"					



5.9 DOC_TVL

DOC_TVL		fnc
Description	Get or set a value in a cell table	
Related	DOC_TINF	
Remark	For column is possible to use Letter notation (A,B.. Aa). With £BLOCK VALtype you can set/get the whole content in ODF	

1.DOC_TVL					
Parameter	Type	Values	Comment	Default	Opt
TABLEptr	PTR		PTR to Table		
COL	symb		Column number (1bsd) or identifier in Letter notation		
ROW	NV		Row 1bsd		
VALtype	symb	£STRING;£FLOAT;£BLOCK(in ODF)	£STRING;£FLOAT;£BLOCK		YES
VALUE	symb		value to set		YES
HEADER	symb	£TRUE;£FALSE	if TRUE col and row refer to header	FALSE	YES
Return					
VALUE	symb				
OnError					
Example					
tblPtr is a pointer to a sheet of a datasheet file. DOC_TVL(tblPtr,£U,9,"STRING",t1); >> Write the content of the attrib t1 in the cell U9 of datasheet as a string. VALtype "STRING" is the best working in all the situation.					

5.10 DOC_TINF

DOC_TINF		fnc
Description	Retrieve information about the table	
Related	DOC_TVL	
Remark		

1.DOC_TINF					
Parameter	Type	Values	Comment	Default	Opt
TABLEptr	PTR		PTR to Table		
WHAT	symb	£COL(Col num); £ROW(Total Row num); £ROWDAT(Row data num)			
Return					
VALUE	symb				
OnError					
Example					
DOC_TINF(tbl,£ROW) >> 11, 1 header + 10 data row					
DOC_TINF(tbl,£ROWDAT) >> 10					



5.11 DOC_SAV

DOC_SAV		fnc
Description	Save the modification applied to the document, on disk	
Related		
Remark		

1.DOC_SAV					
Parameter	Type	Values	Comment	Default	Opt
DOCptr	PTR		PTR to Doc		
Return					
OnError					
Example					
DOC_SAV(docP) >>> save document opened with docP pointer					

6 BUF



6.1 BUF_NEW

BUF_NEW		fnc
Description	Create a Buffer	
Related		
Remark		

1.BUF_NEW					
Parameter	Type	Values	Comment	Default	Opt
NumElem	NV		number of element of TYP		
TYP	symb	£U8;£I8;£C8;£U16;£I16;£C16;£U32;£I32;£F32;£U64;£I64;£F64			
Return					
RES	PTR		Pointer to buffer		
OnError					
RES	symb	NULL			
Example					
<pre>a=BUF_NEW(15,£U8); >>> a points to a 15 bytes buffer, with 15 unsigned char values. b=BUF_NEW(10,£F32); >>> b points to a 40 bytes buffer, with 10 float values.</pre>					

6.2 BUF_SMF

BUF_SMF		fnc
Description	operation on TBL semaphore	
Related	DOOR, SMF	
Remark		

1.BUF_SMF		Command on a SMF			
Parameter	Type	Values	Comment	Default	Opt
PTR to LIS	PTR		PTR to BUF		
cmd	symb	£LCKS(lock 1 level); £LCKR (unlock 1 level);£RST (fully unlock);£TRY(try to lock);£OWN(return the thread owning); £WAIT(wait until is Locked) ; £WAITCNT(count of waiting Thr)	Possible command		YES
Return					
Locks	NV		Number of locks for the caller thread		
Status	NV		if cmd N.D		
Thread	NV		Thread ID of the owner		
OnError					
Example					
see and use SMF					



6.3 BUF_BUF

BUF_BUF		fnc
Description	Create view BUF inside another BUF	
Related		
Remark		

1.BUF_BUF					
Parameter	Type	Values	Comment	Default	Opt
MAINBUF	PTR		PTR to main BUF		
STARTIDX	NV		Start index (in byte, 1bsd) inside MAINBUF		
TYPE	symb	£U8;£I8;£C8;£U16;£I16;£C16;£U32;£I32;£F32;£U64;£I64;£F64	if not declared will be the same of MAINBUF		yes
SIZE	NV		in byte, if not declared will be the trailing size of MAINBUF starting from STARTIDX		yes
Return					
RES	PTR		Pointer to buffer		
OnError					
RES	symb	NULL			
Example					
<pre>a=BUF_NEW(10, £U32); b=BUF_BUF(a, 1, £U8); “b” is able to access “a” BUF data as unsigned char.</pre>					

6.4 BUF_VAL

BUF_VAL		fnc
Description	Get or set a BUF element	
Related		
Remark		

1.BUF_VAL					
Parameter	Type	Values	Comment	Default	Opt
BUF	PTR		PTR to BUF		
INDEX	NV		1bsd; if TYPE is NULL this is in TYPE unit else is in BYTE		
VALUE	symb		if defined is SET		yes
TYPE	symb	£U8;£I8;£C8;£U16;£I16;£C16;£U32;£I32;£F32;£U64;£I64;£F64	if not declared BUF type will be used		yes
SWAP	symb	£BYTE;£WORD;£DWORD(Swap Dword);£XDWORD(Exchange WORD)	Swap to apply		yes
Return					
RES	symb		BUF value at index		
OnError					
RES	symb	NULL			
Example					
BUF a contains (192,168,0, 1) as £U32 <pre> b=BUF_BUF(a,1,£U8); BUF_VAL(a,1,IP_address); part1_of_ip = BUF_VAL(b,4); >>> 192 part2_of_ip = BUF_VAL(b,3); >>> 168 part3_of_ip = BUF_VAL(b,2); >>> 0 part4_of_ip = BUF_VAL(b,1); >>> 1 Alternate method: part1_of_ip = BUF_VAL(a,4,NULL,£U8); >>> 192 part2_of_ip = BUF_VAL(a,3,NULL,£U8); >>> 168 part3_of_ip = BUF_VAL(a,2,NULL,£U8); >>> 0 part4_of_ip = BUF_VAL(a,1,NULL,£U8); >>> 1 </pre>					



6.5 BUF_CPY

BUF_CPY		fnc
Description	Copy a BUF into another	
Related		
Remark	if Offsets are outside ranges the function does nothing.	

1.BUF_CPY					
Parameter	Type	Values	Comment	Default	Opt
BUFDST	PTR		Destination BUF		
BUFSRC	PTR		Source BUF		
SIZE	NV		Number of byte to copy; if not defined the lower of dst,src size is used; the same if size if higher then src or dst size		yes
OFFDST	NV		Offset inside destination		yes
OFFSRC	NV		Offset inside source		yes
Return					
RES	PTR		Pointer to DST buffer		
OnError					
RES	symb	NULL			
Example					
<pre>BUF_CPY(bufDst,bufSrc,20,101,1); !! Copy first 20 byte of bufSrc at byte 100 of destination</pre>					

6.6 BUF_CAT

BUF_CAT		fnc
Description	Concatenate two buffer in a new one	
Related		
Remark	The type of the new BUF will be that of BUFone	

1.BUF_CAT					
Parameter	Type	Values	Comment	Default	Opt
BUFone	PTR		First BUF		
BUFtwo	PTR		Second BUF		
Return					
RES	PTR		Pointer to new BUF		
OnError					
RES	symb	NULL			
Example					
<pre>buf1=BUF_NEW(100,£U8); buf2=BUF_NEW(50,£U8); buf3=BUF_CAT(buf1,buf2); BUF_INFO(buf3,£SIZ) >>> 150, buf3 contains the content of buf1 + buf2</pre>					



6.7 BUF_CMP

BUF_CMP		fnc
Description	Compare for equality two BUF	
Related		
Remark	If the offset are out of the related buffer size, the function does nothing.	

1.BUF_CMP					
Parameter	Type	Values	Comment	Default	Opt
BUFone	PTR		Source BUF		
BUFTwo	PTR		Destination BUF		
SIZ	NV		Number of byte to compare; if not defined the lower of one,two size is used; the same if size if higher then one or two size		yes
OFFone	NV		1bsd; offset inside BUF one		yes
OFFtwo	NV		1bsd; offset inside BUF two		yes
Return					
RES	NV	1	if BUFone = BUFTwo		
RES	NV	0	if BUFone <> BUFTwo		
OnError					
RES	sybm	NULL			
Example					
BUF_COM(buf1,buf2,20,1,101) !!! Compare first 20 byte of buffer 1 with 20 byte at 100 of buffer 2					

6.8 BUF_RST

BUF_RST		fnc
Description	Reset a BUF filling with £U8 value	
Related		
Remark		

1.BUF_RST					
Parameter	Type	Values	Comment	Default	Opt
BUF	PTR		Source BUF		
VALUE	NV		Value for fill	0	yes
SIZ	NV		If N.D. BUF size is used		yes
OFF	NV		The Offset	0	YES
Return					
RES	PTR		to BUF		
OnError					
Example					
BUF_RST(buf) !!! Fill buf with 0 BUF_RST(buf,32,20,101) !!! Fill 20 byte at offset 100(0 bsd) with space					



6.9 BUF_INFO

BUF_INFO		fnc
Description	Return information about a BUF	
Related		
Remark		

1.BUF_INFO					
Parameter	Type	Values	Comment	Default	Opt
BUF	PTR		to BUF		
SELECTOR	symb	£SIZ(Size in U8); £TYP(declared type); £NUM(number of element of BUF type)			
Return					
RES	symb		Requested info		
OnError					
RES	symb	NULL			
Example					
<pre>a=BUF_NEW(10,£U32); BUF_INFO(a,£SIZ); >>> 40 BUF_INFO(a,£TYP); >>> £U32 BUF_INFO(a,£NUM); >>> 10</pre>					

6.10 BUF_SYMB

BUF_SYMB		fnc
Description	Get/set a symbol into/from a BUF	
Related		
Remark	BUF is assumed to contains £C8 while VALUE contains £C16 so ASCII<-->Unicode conversion is performed	

1.BUF_SYMB					
Parameter	Type	Values	Comment	Default	Opt
BUF	PTR		to BUF		
OFF	NV		1bsd;	1	yes
SIZE	NV		size in U8,if N.D. is from OFF to end		yes
VALUE	symb		Set BUF with VALUE		yes
ENCODING	symb	£A(ASCII);£U(UNICODE)	Encoding of symbol	£A	yes
Return					
RES	symb		Requested value		
OnError					
RES	symb	NULL			
Example					
BUF_SYMB(buf,1,15,NULL,£A) >> read an ascii string from buf at position 0					
Note: OFF 0 works as OFF 1					



7

FS

7.1 FS_FIND

FS_FIND		fnc
Description	Search within a directory for files. Wildcard are allowed.	
Related		
Remark	if file name does not contains absolute path current directory is assumed	

1.FS_FIND					
Parameter	Type	Values	Comment	Default	Opt
FILEName	symb		file name to search ; if N.D. "*" is used		yes
Return					
RES	PTR		To a LIS		
OnError					
RES	symb	NULL			
Example					
<pre>lis=FS_FIND("c:\tmp*.txt") >> generate lis, a list pointer, with first element ".", and second "..". The filename in list are without path.</pre>					



7.2 FS_INFO

FS_INFO		fnc
Description	Return file or directory info.	
Related		
Remark	Time is in second since 00:00 of January the 1 st 1970.	

1.FS_INFO					
Parameter	Type	Values	Comment	Default	Opt
NAME	symb		file or directory name		
SELECTOR	symb	£SIZ(Size in U8); £TC(creation time); £TA(last access time); £TW(last write time); £TYP(return £DIR or £FIL);£ATT(OS attribute); £TWHTTP(time write HTTP ts)			
Return					
RES	symb		requested info		
OnError					
RES	symb	NULL			
Example					
<pre>FS_INFO("temp.txt",£SIZ) >> size of temp.txt. 0 : file exists but it is empty NULL : file doesn't exists</pre>					

7.3 FS_LOG

FS_LOG		fnc
Description	Save a Text in a file	
Related		
Remark	if file name does not contains the path current directory is assumed	

1.FS_LOG					
Parameter	Type	Values	Comment	Default	Opt
FILEName	symb				
TEXT	symb		Text to save		
MODE	symb	£NEW(file is created); £ADD(text is added to the end)		£NEW	YES
Return					
RES	PTR	1	OK		
OnError					
RES	symb	NULL			
Example					
<pre>r=FS_LOG("myLog.txt","this is a test"); !!new file or overwrite existing; r=FS_LOG("myLog.txt","this is a test", £ADD); !!the text is added to the end;</pre> <p>A CRLF is inserted Before each inserction.</p>					



7.4 FS_DCWD

FS_DCWD		fnc
Description	Set or get Current Working Directory	
Related		
Remark		

1.FS_DCWD					
Parameter	Type	Values	Comment	Default	Opt
DIRECTORY	symb		(set) new directory		opt
Return					
RES	symb		Corrent Working Directory		
OnError					
RES	symb	NULL			
Example					
FS_DCWD	>> "P:\PWK-PRG\Apps\PWK\PWK-ISP " example of working directory				

7.5 FS_DMK

FS_DMK		fnc
Description	Directory: Create	
Related		
Remark		

1.FS_DMK					
Parameter	Type	Values	Comment	Default	Opt
DIRECTORY	symb		directory name		
Return					
RES	NV	1	Success		
OnError					
Example					
<pre>FS_DMK("C:\PWK-PRG\LOG") >>> 1 success (create LOG but C:\PWK-PRG has to exist) FS_DMK("C:\PWK-PRG\A\B") >>> 0, it doesn't work if C:\PWK-PRG\A doesn't exist</pre>					



7.6 FS_DRM

FS_DRM		fnc
Description	Directory: remove (also sub directory)	
Related		
Remark		

1.FS_DRM					
Parameter	Type	Values	Comment	Default	Opt
DIRECTORY	symb		Directory name		
Return					
RES	NV	1	Success		
OnError					
Example					
FS_DMK("C:\PWK-PRG\A"); >>> C:\PWK-PRG has to exist					
FS_DMK("C:\PWK-PRG\A\B");					
FS_DMK("C:\PWK-PRG\A\B\C");					
FS_DMK("C:\PWK-PRG\A"); >>> 1, delete A, A\B and A\B\C					

7.7 FS_DRN

FS_DRN		fnc
Description	Directory: rename	
Related		
Remark		

1.FS_DRN					
Parameter	Type	Values	Comment	Default	Opt
OLDname	symb		Old directory name		
NEWname	symb		New directory name		
Return					
RES	NV	1	Success		
OnError					
Example					
FS_DRN("C:\PWK-PRG\A", "C:\PWK-PRG\B") !!! Rename A in B					



7.8 FS_FDEL

FS_FDEL		fnc
Description	File: delete	
Related		
Remark		

1.FS_FDEL					
Parameter	Type	Values	Comment	Default	Opt
FILENAME	symb		File name to delete		
Return					
RES	NV	1	Success		
OnError					
Example					

7.9 FS_FCPY

FS_FCPY		fnc
Description	File: copy	
Related		
Remark		

1.FS_FCPY					
Parameter	Type	Values	Comment	Default	Opt
FILEsrc	symb		Source file name		
FILEdst	symb		Destination file name		
Return					
RES	NV	1	Success		
OnError					
Example					



7.10 FS_FREN

FS_FREN		fnc
Description	File: rename	
Related		
Remark		

1.FS_FREN					
Parameter	Type	Values	Comment	Default	Opt
OLDFILENAME	symb				
NEWFILENAME	symb				
Return					
RES	NV	1	Success		
OnError					
RES	symb	NULL			
Example					

7.11 FS_FOPN

FS_FOPN		fnc
Description	File: open unbuffered	
Related	FS_FOPNB	
Remark		

1.FS_FOPN					
Parameter	Type	Values	Comment	Default	Opt
FILENAME	symb		File name		
OPENFLAG	enum	"APPEND, BINARY, CREAT, RDONLY, WRONLY, RDWR, TEXT, TRUNC, EXCL"(all possible flags); £APPEND(open file for append);£BINARY(open binary file);£CREAT(create file);£RDONLY(read only);£WRONLY(write only);£RDWR(read and write);£TEXT(text file);£TRUNC; £EXCL(exclusive access)			
CREATEMODE	enum	£;£READ; £WRITE;"READ,WRITE"			yes
LOCK	symb	£LCKNO;£LCKRW; £LCKR;£LCKW		£LCKNO	YES
Return					
RES	PTR		PTR to open/created file		
OnError					
RES	symb	NULL			
Example					
Prefer BINARY mode and RDWR, for maximal compatibility.					



7.12 FS_FOPNB

FS_FOPNB		fnc
Description	File: open buffered	
Related	FS_FOPNB	
Remark		

1.FS_FOPNB					
Parameter	Type	Values	Comment	Default	Opt
FILENAME	symb		File name		
MODE	enum	£r(read must exists); £w(write if exist its contents are destroyed); £a(append);"r+"(read and write must exists); "w+"(read and write contents are destroyed); "a+"(reading and appending created in not exists); £wb(binary);"wb+"(binary)	Add b=binary t=text		
Return					
RES	PTR		PTR to open/created file		
OnError					
RES	symb	NULL			
Example					
This management is faster, but the writes may not be instantaneous because they are buffered					

7.13 FS_FINP

FS_FINP		fnc
Description	File: BUF input	
Related		
Remark		

1.FS_FINP					
Parameter	Type	Values	Comment	Default	Opt
FILE	PTR		Pointer to an open file		
SIZE	NV		the number of U8 to read		
FILEPOS	symb		NV position in byte (0bsd) or £CUR (current).		
BUF	PTR		Pointer to destination buffer		
Return					
RES	NV		Number of read byte		
OnError					
RES	symb	NULL			
Example					
<pre>FS_FINP(filPtr,100,0,BUF) >> 100 : read 100 byte from file to buffer >> NULL : probably problem with file pointer</pre>					



7.14 FS_FOUT

FS_FOUT		fnc
Description	File: BUF Output	
Related		
Remark		

1.FS_FOUT					
Parameter	Type	Values	Comment	Default	Opt
FILE	PTR		Pointer to an open file		
BUF	PTR		Pointer to source buffer		
FILEPOS	symb		NV position in byte (0bsd) or £CUR (current).		yes
SIZE	NV		the number of U8 to write		yes
Return					
RES	NV		Number of written byte		
OnError					
RES	symb	NULL			
Example					
<pre>FS_FOUT(filPtr,BUF,0,20) >> 20 : write 20 byte from buffer to file >> NULL : probably problem with file pointer</pre>					

7.15 FS_FWRS

FS_FWRS		fnc
Description	File: write symbol	
Related		
Remark		

1.FS_FWRS					
Parameter	Type	Values	Comment	Default	Opt
FILE	PTR		Pointer to an open file		
FILEPOS	symb		NV position in byte (0bsd) or £CUR (current).		
ENCODE	symb	£A(encoding ASCII); £U(encoding UNICODE)	Encoding of VALUES		
VALUES	slis		Symbols to concatenate in a single write		
Return					
RES	NV		Number of written byte		
OnError					
RES	symb	NULL			
Example					
<pre>FS_FWRS(filPtr,0,£A,"Prova prova");</pre>					
<pre>>>> 11 : write string in file >>> NULL : probably a problem with file pointer</pre>					



7.16 FS_FRDS

FS_FRDS		fnc
Description	File: read symbol	
Related		
Remark		

1.FS_FRDS					
Parameter	Type	Values	Comment	Default	Opt
FILE	PTR		Pointer to an open file		
FILEPOS	symb		NV position in byte (0bsd) or £CUR (current).		
ENCODE	symb	£A(encoding ASCII);£U (encoding UNICODE)	Encoding of VALUES		
TERM_SIZE	symb		if NV TERM contains the size (U8), else is the line terminator		
FLAG	symb	£ZEROTOSP	Only for SIZE enc £A or £U, zero are replaced by spaces		YES
Return					
RES	symb		Symbol read (without TERM)		
OnError					
RES	symb	NULL			
Example					
<pre>FS_FWRS(filPtr,0,£A,"Provaccia!"); FS_FWRS(filPtr,0,£A,"Prova"); >>> 5 : write Prova + terminator 0 FS_FRDS(filPtr,0,£A,10) >>> "Prova" FS_FRDS(filPtr,0,£A,10,£ZEROTOSP) >>> "Prova ccia!"</pre>					

7.17 FS_FWRS_F

FS_FWRS_F		fnc
Description	write a text directly to file.	
Related	FS_FRDS_F, FS_FWRS, FS_FRDS,	
Remark	if file name does not contains the path current directory is assumed	

1.FS_FWRS_F					
Parameter	Type	Values	Comment	Default	Opt
FILEName	symb				
TEXT	symb		textto save		
MODE	symb	£NEW(file is created); £ADD(text is added to the end)		£NEW	YES
CODING	symb	£U;£A;£UTF		£U	YES
Return					
RES	NV		Number ob byte written		
OnError					
RES	symb	NULL			
Example					
<pre>r=FS_FWRS_F("myFile.txt","this is a test"); !!new file or overwrite existing; r=FS_FWRS_F("myFile.txt","this is a test", £ADD); !!the text is added to the end;</pre>					
A CRLF is inserted Before each insertion.					



7.18 FS_FRDS_F

FS_FRDS_F		fnc
Description	Read a text directly from file.	
Related	FS_FWRS_F, FS_FWRS, FS_FRDS,	
Remark	if file name does not contains the path current directory is assumed	

1.FS_FRDS_F					
Parameter	Type	Values	Comment	Default	Opt
FILEName	symb				
CODING	symb	£U;£A;£UTF		£U	YES
Return					
RES	symb		the read text		
OnError					
RES	symb	NULL			
Example					
r=FS_FRDS_F("myFile.txt");					

7.19 FS_FPOS

FS_FPOS		fnc
Description	File: Set or Get the read/write current position into a FILE	
Related		
Remark		

1.FS_FPOS					
Parameter	Type	Values	Comment	Default	Opt
FILE	PTR		Pointer to an open file		
FILEPOS	symb		(set) NV position in byte (0bsd) or £CUR (current).		yes
Return					
RES	NV		Position into the file		
OnError					
RES	symb	NULL			
Example					



8

SOK

8.1 SOK_NEW

SOK_NEW		fnc
Description	Create a new SOK (socket)	
Related		
Remark		

1.SOK_NEW					
Parameter	Type	Values	Comment	Default	Opt
TYPE	symb	£UDP;£TCP	SOK type		
IA	symb		Local Internet Address.If N.D. assigned first network device address is used		yes
PORT	NV		Local Port Number (1-65535). if N.D. is taken from the free port pool.		yes
Return					
RES	PTR		To SOK		
OnError					
RES	symb	NULL			
Example					
<pre>SOK_NEW (£TCP) >>> pointer, open a TCP socket SOK_NEW (£TCP,0,6000); >>> pointer, open a TCP socket on port 6000</pre>					



8.2 SOK_LKO

SOK_LKO		fnc
Description	Open a link	
Related		
Remark	This function is used for TCP active connection and in UDP for enable receive	

1.SOK_LKO					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		Pointer to a valid SOK		
LocPORT	NV		if 0 or NULL is assumed to be the same of SOK_NEW		yes
RmtAddres	symb		Remote IA or name (opt for UDP)		yes
RmtPORT	NV		Remote port [1-65535] (opt for UDP)		yes
Return					
RES	NV	>0	Success		
OnError					
RES	symb	ERR			
Example					
<pre>sok= SOK_NEW(£TCP); SOK_LKO(sok,NULL,"192.168.2.200",6000) >>> 1, open a TCP communication with a remote server on port 6000 >>> 0, a problem occurred, manage it</pre>					

8.3 SOK_LKC

SOK_LKC		fnc
Description	Close a link	
Related		
Remark	This function is for TCP connection	

1.SOK_LKC					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		Pointer to a valid SOK		
Return					
RES	PTR		to SOK		
OnError					
RES	symb	NULL			
Example					
SOK_LKC(sok) >>> close socket					



8.4 SOK_LKW

SOK_LKW		fnc
Description	Wait for a link	
Related		
Remark	<p>if TRIG is defined this function return only if there is an error or the SOK is closed.</p> <p>if SELECTOR=£THREAD a new TEMPORARY SOK is create to sustain the conversation with the connecting partner, in a new thread. This SOK is delete at the end of the TRIG. In this way many partner can be served.</p> <p>if SELECTOR=£INPLACE the TRIG is executed as a simple call to an EXO.</p> <p>The TRIG is to set with this parameters:</p> <p>1) SOK, where it's returned PTR to SOK</p> <p>2) ADDRESS, where it's returned the address of incoming transmission</p> <p>3) PORT, where it's returned the port of incoming transmission</p>	

1.SOK_LKW					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
LocPORT	NV		if 0 or NULL is assumed to be the same of SOK_NEW	0	yes
RmtADDRESS	symb		listen only from this address. if N.D. or 0 from all;	NULL	yes
RmtPORT	symb		listen only from this port. if N.D. or 0 from all;	0	yes
TRIG	PTR		Trig to activate at connection	NULL	yes
SELECTOR	symb	£THREAD;£INPLACE	If TRIG is valid it is start in a new Thread or in the actual thread (only if TRIG is valid)	£INPLACE	yes
Return					
RES	NV	1	success		
OnError					
RES	symb	NULL			
Example					

1.SOK_LKW	
<pre>INPLACE VERSION : SOK_LKW(sok,4000,"192.168.2.200",0,NULL,&INPLACE) >>> Awaiting something on port 4000 from a specific remote PC. The program stops on this line, awaiting for data. When something arrives, the program proceed. TRIGGERED VERSION : !!! Define a trigger to manage the remote connections ; trg= TRIG("\NET\Trig"); TRIGSET(trg,"SOK","SOK"); TRIGSET(trg,"ADDRESS","ADDRESS"); TRIGSET(trg,"PORT","PORT"); !!! Link wait trigger version ; !!! To leave this row, something external have to trash sok pointer ; SOK_LKW(sok,0,0,0,trg,&THREAD);</pre>	



8.5 SOK_RCV

SOK_RCV		fnc
Description	Receive data into a BUF	
Related		
Remark		

1.SOK_RCV					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
BUF	PTR		BUF to store received data		
REQDATASIZE	NV		if defined the function wait until the data size is reached.		yes
TMO	NV		Time out in ms		yes
MODE	symb	NULL;£LINE	if £LINE wait for a CRLF (not returned in data)		yes
BUF_IDX	NV		start position in buffer for read values	1	yes
Return					
RES	NV		Number of bytes received		
OnError					
RES	symb	NULL			
Example					
<pre>SOK_RCV(sok,buf,100,10000) >>> number of received bytes. The program stops on this line until it receives 100 bytes or 10 seconds pass. If the timeout expires, the bytes received up to that moment will be present in the buf (less than 100)</pre>					

8.6 SOK_RCVPKT

SOK_RCVPKT		fnc
Description	Receive data into a new BUF	
Related		
Remark		

1.SOK_RCVPKT					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
TMO	NV		Time out in ms		yes
Return					
BUF	PTR		if >0 the PTR of the new buffer containing received data		
OnError					
RES	symb	NULL			
Example					



8.7 SOK_INQ

SOK_INQ		fnc
Description	Inquire to see if any data is arrived	
Related		
Remark		

1.SOK_INQ					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
TMO	NV		Time out in ms		yes
Return					
RES	NV		received packet length		
OnError					
RES	symb	NULL			
Example					
<pre>SOK_INQ(sok,1000) >>> NULL : connection closed >>> 0 : no data received >>> >0 : size of data received</pre>					

8.8 SOK_ADR

SOK_ADR		fnc
Description	Return the address of the last communication peer	
Related		
Remark		

1.SOK_ADR					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
Return					
RES	symb		Internet Address		
OnError					
RES	symb	NULL			
Example					
SOK_ADR(sok) >>> "192.168.2.144"					



8.9 SOK_POR

SOK_POR		fnc
Description	Return the port of the last communication peer	
Related		
Remark		

1.SOK_POR					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
Return					
RES	NV		Port number		
OnError					
RES	symb	NULL			
Example					
SOK_POR(sok) >>> 50132, communication port					

8.10 SOK_STS

SOK_STS		fnc
Description	SOK Status	
Related		
Remark		

1.SOK_STS					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
Return					
RES	NV	0	Closed		
RES	NV	1	Open		
OnError					
RES	symb	NULL			
Example					
<pre>SOK_STS(sok) >>> 0 : communication closed >>> 1 : communication OK</pre>					



8.11 SOK_SND

SOK_SND		fnc
Description	Send data on a SOK from a BUF	
Related	RCVPKT	
Remark		

1.SOK_SND					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
BUF	PTR		To a valid BUF		
SIZE	NV		if N.D. BUF size is used		yes
RmtADDRESS	symb		Remote IA or Name, for TCP connection is unnecessary		yes
RmtPORT	NV		Remote Port for TCP connection is unnecessary		yes
SNDTYPE	symb	NULL;£PKT	Use RCVPKT on the receive side		yes
BUF_IDX	NV		start position in buffer for read values	1	yes
Return					
RES	NV		number of sent bytes		
OnError					
RES	symb	NULL			
Example					
SOK_SND(sok,buf,100,NULL,NULL,£PKT) >>> 100, sent bytes					

8.12 SOK_WRS

SOK_WRS		fnc
Description	Send symbol on a SOK	
Related		
Remark		

1.SOK_WRS					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
DATA	symb		Symb to sent		
ENCODING	symb	£A(ASCII);£UTF;£U(U16); £PKT	Encoding	£PKT	yes
RmtADDRESS	symb		Remote IA or Name, for TCP connection is unnecessary		yes
RmtPORT	NV		Remote Port for TCP connection is unnecessary		yes
Return					
RES	NV		number of sent bytes		
OnError					
RES	symb	NULL			
Example					
<pre>t1="pippo"; SOK_WRS(sok,t1,£A) >>> 5, sent bytes</pre>					



8.13 SOK_RDS

SOK_RDS		fnc
Description	Receive a symbol on a SOK	
Related		
Remark		

1.SOK_RDS					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
ENCODING	symb	£A(ASCII);£UTF;£U(U16); £PKT(datasize and mode are ignored)	Encoding	£PKT	yes
REQDATASIZE	NV		If defined the function will wait this data size ammount		yes
TMO	NV		Time out in ms		yes
MODE	symb	NULL;£LINE	if £LINE wait for a CRLF (not returned in data)		yes
Return					
RES	symb		Received symbol		
OnError					
RES	symb	NULL	err or no bytes received		
Example					
<pre>SOK_RDS(sok,£PKT,NULL,10000) >>> read value >>> NULL, sending error or timeout</pre>					

8.14 SOK_INF

SOK_INF		fnc
Description	Get the dotted IP;PORT of the actual socket or NULL (if not bind)	
Related		
Remark		

1.SOK_INF					
Parameter	Type	Values	Comment	Default	Opt
SOK	PTR		To a valid SOK		
Return					
RES	NV		IP;PORT		
OnError					
RES	symb	NULL			
Example					
SOK_INF(sok) >>> "192.168.2.144;60123"					



9

IEP

9.1 IEP_SRV

IEP_SRV		fnc
Description	Create an IEP SRV (server)	
Related		
Remark		

1.IEP_SRV					
Parameter	Type	Values	Comment	Default	Opt
LocADDRESS	sym		Local IA		
LocPORT	NV		Local port (usually 1800,502 modbus)		
TYPE	sym	£IEP;£MODBUS		£IEP	YES
Return					
RES	PTR		To IEP SRV		
OnError					
RES	symb	NULL			
Example					
iep=IEP_SRV("192.168.2.144",1800,£IEP);					



9.2 IEP_SRVADD

IEP_SRVADD		fnc
Description	Create the reference to a memory area (BUF) in a IEP SRV instance.	
Related		
Remark	Trash IEP DB pointer before trash the IEP SRV Ptr.	

1.IEP_SRVADD					
Parameter	Type	Values	Comment	Default	Opt
IEPSRV	PTR		Valid PTR to IEP SRV		
UNIT	NV		The unit number (for remote reference)		
SUB	NV		The sub unit number (for remote reference – not used if MODBUS)		
BUF	PTR		Valid pointer to a BUF		
Return					
RES	PTR		to a created IEP_DB		
OnError					
RES	symp	NULL			
Example					
<pre>buf1=BUF_NEW(1000,£U8); IEP_SRVADD(iep,100,0,buf1);</pre>					

9.3 IEP_SRVCMD

IEP_SRVCMD		fnc
Description	Start and Stop IEP_SRV	
Related		
Remark		

1.IEP_SRVCMD					
Parameter	Type	Values	Comment	Default	Opt
IEPSRV	PTR		Valid PTR to IEP SRV		
COMMAND	symb	£START;£STOP			
Return					
RES	PTR		to IEP SRV		
OnError					
RES	symb	NULL			
Example					
IEP_SRVCMD (iep, £START) ;					



9.4 IEP_CLI

IEP_CLI		fnc
Description	Create a IEP CLI (client) and connect it to a IEP SRV DB.	
Related		
Remark		

1.IEP_CLI					
Parameter	Type	Values	Comment	Default	Opt
SRVADDRESS	symb		IA of IEP SRV		
SRVPORT	NV		IEP SRV PORT		
CONTYPE	symb	£IEP(XPOLYPLUS, Siemens);£NTLK(Net-Link); £MBUS(modBus)			
CLIADDRESS	symb		Local IA		yes
CLIPORT	NV		Local Port		yes
Return					
RES	PTR		IEP CLI pointer		
OnError					
RES	symb	NULL			
Example					
iep=IEP_CLI("192.168.2.144",1800,£IEP);					

9.5 IEP_CLICON

IEP_CLICON		fnc
Description	Set IEP CLI connection parameters .	
Related		
Remark		

1.IEP_CLICON					
Parameter	Type	Values	Comment	Default	Opt
IEP_CLI	PTR		Pointer to a valid IEP CLI		
UNIT	NV		Unit number of IEP SRV DB		
SUB	NV		Sub Unit number of IEP SRV DB		
SWAP	enum	NULL;"HEAD,WORD";"HEAD,DWORD";"WORD";"DWORD"	Swap of head and data		yes
MTU	NV		Maximum Transfer Unit		yes
Return					
RES	symb				
OnError					
RES	symb	NULL			
Example					
IEP_CLICON(iep,100,0);					



9.6 IEP_CLIDAT

IEP_CLIDAT		fnc
Description	Set IEP CLI data parameters .	
Related		
Remark		

1.IEP_CLIDAT					
Parameter	Type	Values	Comment	Default	Opt
IEP_CLI	PTR		Pointer to a valid IEP CLI		
PAGE	NV		Page Number (Siemens/NTLK DB)		
OBJ	NV	NULL;131(Siemens merker);132(Siemens DB);49(NTLK db);51(NTLK merker);52(NTLK I/O);53 (NTLK Counters);54(NTLK timers)	NULL for MODBUS		
OBJsize	NV		Size in byte of object		
OBJnum	NV		Number of object		
OFFSET	NV		(0bsb) offset in IEP SRV DB		
OFFSETunit	symb	£BYTE;£WORD; £DWORD	Measure unit of address	£BYTE	YES
Return					
RES	PTR		IEP CLI		
OnError					
RES	symb	NULL			
Example					
IEP_CLIDAT (iep,0,0,1,200,0);					

9.7 IEP_CLIBUF

IEP_CLIBUF		fnc
Description	Set the IEP CLI default BUF for read/write operation.	
Related		
Remark	When you specify a buf if the other condition is not set , is set for both	

1.IEP_CLIBUF					
Parameter	Type	Values	Comment	Default	Opt
IEP_CLI	PTR		valid IEP CLI ptr		
BUF	PTR		valid Buf		
FOR	symb	£READ;£WRITE	you can use two distinct buffer	READ	TES
Return					
RES	PTR		IEP CLI ptr		
OnError					
RES	symb	NULL			
Example					
<pre>buf=BUF_NEW(1000, £U8); IEP_CLIBUF(iep_buf);</pre>					



9.8 IEP_CLIRD

IEP_CLIRD		fnc
Description	Read data from a remote IEP SRV into a BUF	
Related	IEP_CLI_WR	
Remark	Offset is in the same unit of OFFSETunit parameter expressed IEP_CLIDAT	

1.IEP_CLIRD					
Parameter	Type	Values	Comment	Default	Opt
IEP_CLI	PTR		valid IEP CLI ptr		
OFFSET	NV		Offset 1bsd		YES
SIZ	NV		In byte		YES
Return					
RES	symb	>0	Success		
OnError					
RES	symb	ERR			
Example					
res=IEP_CLIRD(iep); >>> Read all the area ;					
res=IEP_CLIRD(iep,1,20); >>> Read only the first part of the area ;					

9.9 IEP_CLIWR

IEP_CLIWR		fnc
Description	Write data from BUF to a remote IEP SRV	
Related	IEP_CLIRD	
Remark	Offset is in the same unit of OFFSETunit parameter expressed IEP_CLIDAT	

1.IEP_CLIWR					
Parameter	Type	Values	Comment	Default	Opt
IEP_CLI	PTR		valid IEP CLI ptr		
OFFSET	NV		Offset 1bsd		YES
SIZ	NV		In byte		YES
Return					
RES	symb	>0	Success		
OnError					
RES	symb	ERR			
Example					
res=IEP_CLIWR(iep); >>>Write all the area ;					
res=IEP_CLIWR(iep,1,20); >>> Write only first 20 bytes of the buffer ;					



10

FTP

10.1 FTP_NEW

FTP_NEW		fnc
Description	Create a Ptr to a FTP Server	
Related		
Remark		

1.FTP_NEW					
Parameter	Type	Values	Comment	Default	Opt
ADDRESS	symb		IA of server		
USER	synb		user name		yes
PASSWORD	symb		password		yes
Return					
RES	PTR		to FTP		
OnError					
RES	symb	NULL			
Example					
ftpP=FTP_NEW("192.168.2.144","xplab","xppsw")					



10.2 FTP_DIR

FTP_DIR		fnc
Description	Set or get remote or local directory	
Related		
Remark		

1.FTP_DIR					
Parameter	Type	Values	Comment	Default	Opt
FTP	PTR		valid pointer to FTP		
DIRECTORY	symb		Directory		yes
SELECTOR	symb	£LOC(lget/set ocal); £RMT(get/set remote);£LST(list remote)			yes
Return					
RES	NV	1	success		
LIS	PTR		if £LST the LIS of the remote directory		
OnError					
RES	symb	NULL			
Example					
FTP_DIR (ftpP, NULL, £LOC) >>> "c:\PWK-PRG\"					

10.3 FTP_GET

FTP_GET		fnc
Description	Copy a remote file to local directory	
Related		
Remark		

1.FTP_GET					
Parameter	Type	Values	Comment	Default	Opt
FTP	PTR		valid pointer to FTP		
FILE	symb		remote file name		
LOCName	symb		the new name on local machine		yes
POSTACTION	symb	NULL;£DEL(delete from remote after get)			
Return					
RES	NV	1	success		
OnError					
RES	symb	NULL			
Example					
<pre>FTP_GET(ftpP, "report.txt", NULL, £DEL) >>> get "report.txt" from server and delete it</pre>					



10.4 FTP_MGET

FTP_MGET		fnc
Description	Copy a multiple remote file to local directory	
Related		
Remark		

1.FTP_MGET					
Parameter	Type	Values	Comment	Default	Opt
FTP	PTR		valid pointer to FTP		
FILE	symb		remote file name (accept wild cards)		
POSTACTION	symb	NULL;£DEL(delete from remote after get)			
Return					
RES	NV	1	success		
OnError					
RES	symb	NULL			
Example					
<pre>FTP_GET(ftpP,"*.txt",£DEL) >>> get all txt files from server and delete them</pre>					

10.5 FTP_PUT

FTP_PUT		fnc
Description	Copy a local file on remote FTP server	
Related		
Remark		

1.FTP_PUT					
Parameter	Type	Values	Comment	Default	Opt
FTP	PTR		valid pointer to FTP		
FILE	symb		local file name		
RMTName	symb		the new name on remote machine		yes
Return					
RES	NV	1	success		
OnError					
RES	symb	NULL			
Example					
<pre>FTP_PUT(ftpP,"setTest.txt","set.txt") >>> copy setTest.txt to server, renaming it "set.txt"</pre>					



10.6 FTP_DEL

FTP_DEL		fnc
Description	Delete a remote file	
Related		
Remark		

1.FTP_DEL					
Parameter	Type	Values	Comment	Default	Opt
FTP	PTR		valid pointer to FTP		
FILE	symb		remote file name (accept wild cards (tbv))		
Return					
RES	NV	1	success		
OnError					
RES	symb	NULL			
Example					
<pre>FTP_DEL(ftpP,"set.txt") >>> delete "set.txt" on the server</pre>					

10.7 FTP_STS

FTP_STS		fnc
Description	Return connection status	
Related		
Remark		

1.FTP_STS					
Parameter	Type	Values	Comment	Default	Opt
FTP	PTR		valid pointer to FTP		
Return					
RES	NV	1	ok		
OnError					
RES	symb	NULL			
Example					
FTP_STS(ftpP) >>> 1 if connected					



10.8 FTP_RECON

FTP_RECON		fnc
Description	Reconnect wit stored parameter	
Related		
Remark		

1.FTP_RECON					
Parameter	Type	Values	Comment	Default	Opt
FTP	PTR		valid pointer to FTP		
Return					
RES	NV	1	Ok		
OnError					
RES	symb	NULL			
Example					
FTP_RECON(ftpP) >>> reconnect to remote server					

11 WEB



11.1 WEB_NEW

WEB_NEW		fnc
Description	Create an WEB server	
Related		
Remark		

1.WEB_NEW					
Parameter	Type	Values	Comment	Default	Opt
ADDRESS	symb		IA on which belongs the server		
PORT	NV		Usually 80		
DIRECTORY	symb		WEB root directory		
NAME	symb		WEB name		
Return					
RES	PTR		WEB pointer to created service		
OnError					
RES	symb	NULL			
Example					

11.2 WEB_SET

WEB_SET		fnc
Description	Set TRIG Parameters for manage client request	
Related		
Remark	Only GET and POST are implemented. TRIG parameters are for TRIG CALL: WEB (PTR); SOK (PTR); rqsMth; rqsUrl; rqsVer; rqsQry (LIS PTR); KB1 . TRIG return in rqsRes its reply.	

1.WEB_SET					
Parameter	Type	Values	Comment	Default	Opt
WEB	PTR		to a valid WEB		
COMMAND	symb	£GET;£POST; £HEAD; £OPTIONS;£PUT; £DELETE;£TRACE; £CONNECT	Command to which associate the TRIG		
TRIG	PTR		The TRIG to activate		
Return					
OnError					
Example					
KB1 contains a CONPAR item and if a query is detected a QRY sub item in this case, the also rqsQry contains the LIS of the query parameters.					



11.3 WEB_CMD

WEB_CMD		fnc
Description	Command WEB	
Related		
Remark		

1.WEB_CMD					
Parameter	Type	Values	Comment	Default	Opt
WEB	PTR		to a valid WEB		
COMMAND	symb	£START;£STOP			
Return					
OnError					
Example					

11.4 WEB_FIL

WEB_FIL		fnc
Description	Send a file	
Related	WEB_TXT WEB_BUF	
Remark	To be used inside a TRIG	

1.WEB_FIL					
Parameter	Type	Values	Comment	Default	Opt
WEB	PTR		to a valid WEB		
SOK	PTR		PTR to SOK		
FILENAME	symb		File to send		
WithName	symb		The name to show instead of Filename		YES
HEADERINFO	symb		To put in the header		YES
Return					
RES	NV	1	OK		
OnError					
RES	symb	ERR			
Example					



11.5 WEB_TXT

WEB_TXT		fnc
Description	Send a Text	
Related	WEB_FIL WEB_BUF	
Remark	To be used inside a TRIG	

1.WEB_TXT					
Parameter	Type	Values	Comment	Default	Opt
WEB	PTR		to a valid WEB		
SOK	PTR		PTR to SOK		
FILENAME	symb		a name with dot extension		
TEXT	symb		The text to send		
HEADERINFO	symb		To put in the header		YES
Return					
RES	NV	1	OK		
OnError					
RES	symb	ERR			
Example					

11.6 WEB_BUF

WEB_BUF		fnc
Description	Send a Text	
Related	WEB_FIL WEB_TXT	
Remark	To be used inside a TRIG	

1.WEB_BUF					
Parameter	Type	Values	Comment	Default	Opt
WEB	PTR		to a valid WEB		
SOK	PTR		PTR to SOK		
FILENAME	symb		a name with dot extension		
BUFPTR	symb		a PTR to a buf		
HEADERINFO	symb		To put in the header		YES
Return					
RES	NV	1	OK		
OnError					
RES	symb	ERR			
Example					



11.7 WEB_RPL

WEB_RPL		fnc
Description	Send an WEB/1.1 reply	
Related		
Remark	in the form of : WEB/1.1 code text CRLF trailer CRLF	

1.WEB_RPL					
Parameter	Type	Values	Comment	Default	Opt
WEB	PTR		to a valid WEB		
SOK	PTR		PTR to SOK		
CODE	NV		WEB/1.1 reply code		
TEXT	symb		Reply text		
TRAILER	symb		Reply Trailer		
Return					
Example					

12 DT



12.1 DT_TIME

DT_TIME		fnc
Description	Get or set Sys or Loc Time	
Related		
Remark	Setting the time can be limited by Windows, in many case you must start the program with administrator right.	

1.DT_TIME		Set or get Date and Time			
Parameter	Type	Values	Comment	Default	Opt
WHAT	symb	£LOC;£SYS		LOC	
YEAR	NV				yes
MONTH	NV		(1-12)		yes
DAY	NV		(1-31)		yes
HOURL	NV		(0-23)		yes
MINUTE	NV		(0-59)		yes
SECOND	NV		(0-59)		yes
Return					
RES	symb		TS		
OnError					
RES	symb	NULL			
Example					
DT_TIME (£LOC) ; ==> 2016-06-22 09:10:01					

12.2 DT_TS

DT_TS		fnc
Description	Generate a Time Stamp	
Related		
Remark	Base for TS is January 1 st 00:00:00 1970. Ifni called without parameters return NOW TS	

1.DT_TS					
Parameter	Type	Values	Comment	Default	Opt
YEAR	NV				yes
MONTH	NV		(1-12)		yes
DAY	NV		(1-31)		yes
HOUR	NV		(0-23)		yes
MINUTE	NV		(0-59)		yes
SECOND	NV		(0-59)		yes
DAYLIGHT	NV	-1;0;1	Day light selector	0	yes
Return					
RES	NV		TS		
OnError					
RES	symb	NULL			
Example					
DAYLIGHT selector: - Zero (0) to indicate that standard time is in effect. - A value greater than 0 to indicate that daylight saving time is in effect. - A value less than zero to have the C run-time library code compute whether standard time or daylight saving time is in effect.					



12.3 DT_TSDEC

DT_TSDEC		fnc
Description	Decode a TS (time stamp), or set CLOCK OFFset	
Related		
Remark	if TS is not provided CLOCK is used,	

1.DT_TSDEC					
Parameter	Type	Values	Comment	Default	Opt
TS	NV			CLOCK	YES
SELECTOR	symb	£DD(day); £MM(month); £YY(year); £HH(hour); £MN(minute); £SS(second); £DW(day of week); £YD(day of the year); £HTTP(HTTP Time stamp); £ALL(date and time)			
TYPE		£LOC; £LOCSLR(Solar); £ABS	LOCal or ABSolute	£LOC	YES
Return					
RES	NV		decoded value		
DT	symb		if Selector=£ALL Date and Time		
OnError					
RES	symb	NULL			
Example					
<pre>DT_TSDEC (CLOCK, £YY) >>> 2020 DT_TSDEC (CLOCK, £HTTP) >>> "Wed, 03 Jun 2020 08:45:31 GMT"</pre>					

2.DT_TSDEC		Set the CLOCK OFFset			
Parameter	Type	Values	Comment	Default	Opt
DELTA	NV		OFFset in second		
SELECTOR	symb	£OFF			
Return					
RES	symb		LOCAl Date and time String using CLOCK		
OnError					
RES	symb	NULL			
Example					
<pre>CLOCK; >>> 159127699 DT_TSDEC (-10, £OFF); !!! After 10 seconds since the first clock; CLOCK; >>> 159127699 (now clock gives values decreased by 10)</pre>					

13 /*FREE*/



14 GUI

14.1 GUI_OPN

GUI_OPN		fnc
Description	Create or Load a GUI	
Related	GUI_SRV	
Remark	From a GUI item (to be passed as address) or from a .pki file	

1.GUI_OPN					
Parameter	Type	Values	Comment	Default	Opt
GUIREF	symb		Address (\$) of a GUI item [or .pki file name (*)]		
IA	symb		IA	127.0.0.1	yes
PORT	NV		PORT	1956	yes
Return					
RES	PTR		Pointer to GUI		
OnError					
RES	symb	NULL			
Example					
<p>\gMain is a GUI with _AUTOLOAD_="OFF"</p> <p>There has to be a GUI.EXE active at ip and port specified in the example (1956 is the local default port for GUI).</p> <p>GUI_OPN(\$\gMain,"192.168.2.144",1956).</p> <p>_PTR_ in \gMain will be set to pointer. Before load you can edit _FILE_ in order to load a specified page.</p> <p>* NOTE</p> <p>It's possible to load a page using GUIREF = "pki file name". It generates a CLI pointer to the page, that you can use to show or hide the loaded page. But it is not possible to reference to trig or to internal data.</p> <p>A page opened in this way can be useful to take advantage of the send and receive functionality of the files, not to burden the interaction with the other existing GUIs</p>					



14.2 GUI_ALV

GUI_ALV		fnc
Description	Check if the connection is alive	
Related		
Remark		

1.GUI_ALV					
Parameter	Type	Values	Comment	Default	Opt
GUIREF	symb		Address (\$) of a GUI item or .pki file name		
TMO	NV		ms time out	1000	yes
Return					
RES	NV	1			
RES	NV	-2	Time Out		
OnError					
RES	symb	NULL			
Example					
GUI_ALV(_PTR_@gMain,1000) >> 1 : ok, communication active >> -2, after 1 second, timeout, GUI is not connected.					

14.3 GUI_SND

GUI_SND		fnc
Description	Send a command to the GUI	
Related		
Remark	FOR EXEC : To use e different separator (;) use _SEP=X where X is the sep to use	

1.GUI_SND					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		valid GUI pointer		
COMMAND	symb	£SHOW;£HIDE; £TOMIN(reduce pag to icon); £TONRM(normalize page from icon); £CLOSE; £DISABLE; £ENABLE;£EXEC, £CURSOR, £SEND_KEY:(see example)			
Return					
RES	symb				
OnError					
RES	symb	NULL			
Example					



1.GUI_SND	
<p>CLOSE ==> Terminate the GUI, unloading it. It needs GUI_OPN to reload it.</p>	
<p>SHELL (remote execution) ----- "EXEC:_OS_SHELL_;action(SHOW OPEN, PRINT, WAIT),parameter,command;\$RplAddr"</p>	
<p>FS_INFO ----- "EXEC:_FS_INFO_;fleDirName,parameter;\$RplAddr" (see FS_INFO for parameters)</p>	
<p>GUI_INFO ----- "EXEC:_GUI_INFO_;command(VDUSIZ);\$RplAddr" VDUSIZ >>> X:%d;Y:%d Return vdu size in pixel</p>	
<p>CURSOR ----- "EXEC:CURSOR; (DEFAULT WAIT)"</p>	
<p>SEND_KEY ----- "EXEC:SEND_KEY; key" Each key is represented by one or more characters. To specify a single keyboard character, use the character itself. For example, to represent the letter A, pass in the string "A" to the method. To represent more than one character, append each additional character to the one preceding it. To represent the letters A, B, and C, specify the parameter as "ABC". The plus sign (+), caret (^), percent sign (%), tilde (~), and parentheses () have special meanings to SendKeys. To specify one of these characters, enclose it within braces ({}). For example, to specify the plus sign, use "{+}". To specify brace characters, use "{}" and "{}". Brackets ([]) have no special meaning to SendKeys, but you must enclose them in braces. BACKSPACE {BACKSPACE}, {BS}, or {BKSP} ;BREAK {BREAK}; CAPS LOCK {CAPSLOCK}; DEL or DELETE {DELETE} or {DEL}; DOWN ARROW {DOWN} ;END {END} ; ENTER {ENTER} or ~; ESC {ESC} ;HELP {HELP} ;HOME {HOME} ;INS or INSERT {INSERT} or {INS} ;LEFT ARROW {LEFT} ; NUM LOCK {NUMLOCK} ;PAGE DOWN {PGDN} ;PAGE UP {PGUP}; PRINT SCREEN {PRTSC} (reserved for future use); RIGHT ARROW {RIGHT} ;SCROLL LOCK {SCROLLLOCK} ;TAB {TAB} ;UP ARROW {UP} ; {F1};{F2} ;{F3} ;{F4} ;{F5} ;{F6} ;{F7} ;{F8} ;{F9} ;{F10} ;{F11} ;{F12} ;{F13} ;{F14} ; {F15}; {F16}; Keypad add {ADD} ;Keypad subtract {SUBTRACT} ;Keypad multiply {MULTIPLY} ;Keypad divide {DIVIDE} ; To specify keys combined with any combination of the SHIFT, CTRL, and ALT keys, precede the key code with one or more of the following codes: SHIFT +; CTRL ^ ;ALT % To specify that any combination of SHIFT, CTRL, and ALT should be held down while several other keys are pressed, enclose the code for those keys in parentheses. For example, to specify to hold down SHIFT while E and C are pressed, use "+(EC)". To specify to hold down SHIFT while E is pressed, followed by C without SHIFT, use "+EC". To specify repeating keys, use the form {key number}. You must put a space between key and number. For example, {LEFT 42} means press the LEFT ARROW key 42 times; {h 10} means press H 10 times.</p>	
<p>CLOSE ----- Close the page, destroying the pointer. To reoper the page, it's necessary to use GUI_OPN</p>	

14.4 GUI_MOUSE

GUI_MOUSE		fnc
Description	Get mouse values	
Related		
Remark		

1.GUI_MOUSE					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		valid GUI pointer		
SELECTOR	symb	£NULL(return change counter); £X ;£Y; £Xw(screen); £Yw(screen); £Xc(control); £Yc(control); £ON(if on the page);£Bl(button left);£BR(Button right)			yes
Return					
RES	symb		requested value		
OnError					
RES	symb	NULL			
Example					
GUI_MOUSE(_PTR_) >>> return a value that changes whenever a mouse interaction involves the page. It can be a click, a movement ...					



14.5 GUI_DUPNEW

GUI_DUPNEW		fnc
Description	Duplicate a GUI	
Related	GUI_DUPNEW	
Remark		

1.GUI_DUPNEW					
Parameter	Type	Values	Comment	Default	Opt
GUIREF	NV		Address (\$) of a GUI item		
NEWGUIMANE	symb		the name of the duplicated GUI		
ALIAS	symb	£NO;£YES	if yes an alias is created		yes
Return					
RES	NV		Address (\$) of the duplicated GUI item		
OnError					
RES	symb	NULL			
Example					
<pre>CREATE ----- newGuiNam="\Calc_ "++conCnt@\Main; guiDupPtr=GUI_DUPNEW(\$(\Calc),newGuiNam,£YES); smf(1,£LCKS) conNum@\Main=conNum@\Main+1; !!Increment reference cont; smf(1,£LCKR) opnPtr=GUI_OPN(guiDupPtr,guiIp,guiPort); !!opnPtr is equivalent to REF("_PTR_@"++newGuiNam); GUI_SND(opnPtr,£SHOW); DESTROY ----- GUI_DUPDEL(guiDupPtr); !! no need to TRASH; SMF(1,£LCKS); conNum@\Main=conNum@\Main-1; SMF(1,£LCKR);</pre>					

14.6 GUI_DUPDEL

GUI_DUPDEL		fnc
Description	Delete a Duplicated GUI	
Related	GUI_DUPNEW	
Remark	The gui is also TRASHed	

1.GUI_DUPDEL					
Parameter	Type	Values	Comment	Default	Opt
GUIREF	NV		Address (\$) of a GUI item		
Return					
RES	NV	1	OK		
OnError					
RES	symb	NULL			
Example					
see GUI_DUPNEW					



14.7 GUI_STREAM

GUI_STREAM		fnc
Description	send a stream image to the gui	
Related	BUF, OCV_MAT	
Remark		

1.GUI_STREAM					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		Valid Pointer to a GUI		
IMAGE_ADDRESS	NV		\$address of a IMAGE item		
SOURCE	PTR		to BUF or ocv MAT		
WHIDE	NV		in pixel		OPT
HEIGHT	NV		in pixel		OPT
LINE	NV		line (row) size in byte		OPT
FORMAT	symb		Image Microsoft Format		OPT
Return					
RES	NV	1	Ok		
OnError					
RES	symb	ERR			
Example					
<pre>r=GUI_STREAM(gui,\$image@MainGui\pag\panel, mat);</pre>					

14.8 GUI_FILWR

GUI_FILWR		fnc
Description	Write a file on the system (PC) of the GUI	
Related	GUI_FILRD	
Remark		

1.GUI_FILWR					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		Valid Pointer to a GUI		
LOCname	symb		Local name		
RMTname	symb		Name at GUI		yes
Return					
RES	NV	1	OK		
OnError					
RES	symb	ERR			
Example					
Copy the file log.txt from PC with PWK software to the PC of the GUI (may be the same), renaming it 2020-05-26.txt:					
GUI_FILWR("c:\archive\log.txt","c:\PWK-TMP\2020-05-26.txt") >>> 1 = success					



14.9 GUI_FILRD

GUI_FILRD		fnc
Description	Read a file on the system (PC) of the GUI	
Related	GUI_FILWR	
Remark		

1.GUI_FILRD					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		Valid Pointer to a GUI		
LOCname	symb		Local name		
RMTname	symb		Name at GUI		yes
Return					
RES	NV	1	OK		
OnError					
RES	symb	ERR			
Example					
Copy the file 2020-05-26.txt from PC of the GUI to the PC with PWK software (may be the same), renaming it log.txt: GUI_FILRD("c:\archive\log.txt", "c:\PWK-TMP\2020-05-26.txt") >>> 1 = success					

14.10 GUI_SRV

GUI_SRV		fnc
Description	Action on gui server	
Related	GUI_OPN	
Remark		

1.GUI_SRV		GUI Server registration for PSV connection			
Parameter	Type	Values	Comment	Default	Opt
ACTION	symb	£REG	Register a Gui Server		
SOCK	PTR		pointer to SOCK		
cbPort	NV		Call Back Port		
dclPort	NV		Declared Port		
Return					
RES	NV	1	OK		
OnError					
RES	symb	ERR			
Example					
When a GUI connect itself to an application and send a GUI@ip;port;PSV message the gui should be registered as a Server provider. After this operation all GUI_OPN on that gui are made automatically in PSV mode: upon request is the gui that open the connection on the cbPort (that should be open on the firewall),					

2.GUI_SRV		GUI Server Gateway start			
Parameter	Type	Values	Comment	Default	Opt
ACTION	symb	£GTW	Start a gateway service		
GUI_IP	I.A.		Gui Ip		
GUI_PORT	NV		Gui port		
tmo	NV		time out for the first connection	35000	OPT
Return					
PORT	NV		Port for client		
OnError					
RES	symb	ERR			
Example					
A Gateway service is started (with tmo limit) for client that want to connect to a remote gui, The service is started at the PORT on the address of this application that are passed to client application. The Gui server has to be registered before.					



14.11 GUI_DLGMSG

GUI_DLGMSG		fnc
Description	open a Message Box	
Related	GUI_SND, GUI_EXCMSG(deprecated)	
Remark		

1.GUI_DLGMSG					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid Gui PTR		
TITOLO	symb		Main text		
TEXT	symb		Message Text		
BUTTON	symb	£OK;£OK_CANC; £YES_NO; £YES_NO_CANC	Shown button		
ICON	symb	£ERR; £EXCLAMATION; £QUESTION; £STOP; £WARNING; £INFO	Dialog icon		
DISABLE	symb	£TRUE;£FALSE		TRUE	YES
Return					
RES	NV		The reply		
OnError					
RES	symb	NULL			
Example					
<p>The dialog appears and is synchronous.</p> <pre>t1=GUI_DLGMSG(<ptr>,"Test","Test text",£OK_CANC,£QUESTION,NULL);</pre> <p>The dialog is shown and the code waits for an input from the dialog. In the example, at the end, t1 will value £OK or £CANCEL or NULL .</p>					

14.12 GUI_DLGOFD

GUI_DLGOFD		fnc
Description	open an Open File Dialog	
Related	GUI_SND, GUI_EXCOFD(deprecated)	
Remark	if multi reply files names are separated by semi colon	

1.GUI_DLGOFD					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid GUI PTR		
MULTISELECT	symb	£TRUE; £FALSE	Allow multi selection ?	FALSE	YES
FILENAME	symb		Proposed file name	NULL	YES
DEF_EXTENTION	symb		default file extention	NULL	YES
INI_DIR	symb		Initial directory	NULL	YES
FILTER	symb		Filter string	All files (.*) *.*	YES
RESTORE_DIR	symb	£TRUE; £FALSE	If after the selection the directory should be restored	TRUE	YES
CHECK_FILE	symb	£TRUE; £FALSE	If file existence should be checked	TRUE	YES
TITLE	symb		Page title		YES
Return					
RES	NV				
OnError					
RES	symb	ERR			
Example					
<pre>fn=GUI_DLGOFD(_PTR_@\MP, £FALSE, NULL, NULL, NULL, "Kb1 *.kb1 All files (*.*) *.*", £FALSE, £TRUE, "test"); ~fn==0 >>> user aborted or error</pre>					



14.13 GUI_DLGSFD

GUI_DLGSFD		fnc
Description	open an SAVE File Dialog	
Related	GUI_SND, GUI_EXCSFD(deprecated)	
Remark		

1.GUI_DLGSFD					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid GUI PTR		
FILENAME	symb		Proposed file name	NULL	YES
DEF_EXTENTION	symb		default file extention	NULL	YES
INI_DIR	symb		Initial directory	NULL	YES
FILTER	symb		Filter String	All files (*.*) *.*	YES
RESTORE_DIR	symb	£TRUE;£FALSE	If after the selection the directory should be restored	TRUE	YES
TITLE	symb		Page title		YES
Return					
RES	NV				
OnError					
RES	symb	NULL			
Example					
<pre>fn=GUI_DLGSFD(_PTR_,"log.txt",NULL,NULL,"Txt files (*.txt) *.txt",£FALSE,"Save file name"); ~fn==0 >>> user aborted or error</pre>					

14.14 GUI_DLGFLD

GUI_DLGFLD		fnc
Description	open an Folder browser Dialog	
Related	GUI_SND, GUI_EXCFLD(deprecated)	
Remark	you must wait for the reply	

1.GUI_DLGFLD					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid GUI PTR		
PATH	symb		Initial Path	NULL	YES
NEWFOLDER	symb	£FALSE;£TRUE	Show New Folder Button	FALSE	YES
DESCRIPTION	symb		Description		YES
Return					
RES	NV				
OnError					
RES	symb	NULL			
Example					
<pre>dir=GUI_DLGFLD(_PTR_,NULL,£FALSE,"Select a folder to save to"); ~dir==0 >>> user aborted or error</pre>					



14.15 GUI_DLGCLR

GUI_DLGCLR		fnc
Description	open a COLOR dialog	
Related	GUI_SND	
Remark		

1.GUI_DLGCLR					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid GUI PTR		
FORMAT	symb	£#(css); £Frgb(function); £RGBA;£name;			
COLOR	symb		Actual color if any		YES
Return					
RES	NV				
OnError					
RES	symb	NULL			
Example					
# like #AA00BB					
Frgb like rgb(20, 80, 255) or rgba(20, 80, 255, 0.6) :					
RGBA like R;G;B;A; 20;40;80 or 20;40;80;255					
name like red, blue , yellow					
if format is non defined but there is a value the format of the value is used.					
The default format is eXadecimal with A component in the high order.					

14.16 GUI_EXCMSG

GUI_EXCMSG		fnc
Description	open a Message Box - DEPRECATED	
Related	GUI_SND, GUI_DLGMSG	
Remark	you must wait for the reply	

1.GUI_EXCMSG					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid Gui PTR		
TITOLO	symb		Main text		
TEXT	symb		Message Text		
BUTTON	symb	£OK; £OK_CANC; £YES_NO; £YES_NO_CANC	Shown button		
ICON	symb	£ERR; £EXCLAMATION; £QUESTION; £STOP; £WARNING; £INFO	Dialog icon		
DISABLE	symb	£TRUE; £FALSE		TRUE	YES
\$ADDRESS	symb		where put the Reply		YES
Return					
RES	NV				
OnError					
RES	symb	ERR			
Example					
<p>The dialog appears and is asynchronous. To wait it or to catch the pressed button key:</p> <pre>(t1 is a local attribute) t1=NULL; GUI_EXCMSG(<ptr>,"Test","Test text",£OK_CANC,£QUESTION,NULL,\$t1); WAITCND(\$t1,£NOTNULL);</pre> <p>The dialog is shown and the code waits for an input from the dialog. In the example, at the end, t1 will value £OK or £CANCEL.</p>					



14.17 GUI_EXCOFD

GUI_EXCOFD		fnc
Description	open an Open File Dialog - DEPRECATED	
Related	GUI_SND, GUI_DLGOFD	
Remark	you must wait for the reply, if multi reply files names are separated by semi colon	

1.GUI_EXCOFD					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid GUI PTR		
MULTISELECT	symb	£TRUE;£FALSE	Allow multi selection ?	FALSE	YES
FILENAME	symb		Proposed file name	NULL	YES
DEF_EXTENTION	symb		default file extention	NULL	YES
INI_DIR	symb		Initial directory	NULL	YES
FILTER	symb		Filter string	All files (.*) *.*	YES
RESTORE_DIR	symb	£TRUE;£FALSE	If after the selection the directory should be restored	TRUE	YES
CHECK_FILE	symb	£TRUE;£FALSE	If file existence should be checked	TRUE	YES
ADDRESS	symb		where put the Reply (\$)		YES
TITLE	symb		Page title		YES
Return					
RES	NV				
OnError					
RES	symb	ERR			
Example					

14.18 GUI_EXCSFD

GUI_EXCSFD		fnc
Description	open an SAVE File Dialog - DEPRECATED	
Related	GUI_SND, GUI_DLGSFD	
Remark	you must wait for the reply	

1.GUI_EXCSFD					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid GUI PTR		
FILENAME	symb		Proposed file name	NULL	YES
DEF_EXTENTION	symb		default file extention	NULL	YES
INI_DIR	symb		Initial directory	NULL	YES
FILTER	symb		Filter String	All files (.*) *.*	YES
RESTORE_DIR	symb	£TRUE;£FALSE	If after the selection the directory should be restored	TRUE	YES
ADDRESS	symb		where put the Reply (\$)		YES
TITLE	symb		Page title		YES
Return					
RES	NV				
OnError					
RES	symb	ERR			
Example					



14.19 GUI_EXCFLD

GUI_EXCFLD		fnc
Description	open an Folder browser Dialog - DEPRECATED	
Related	GUI_SND, GUI_DLGFLD	
Remark	you must wait for the reply	

1.GUI_EXCFLD					
Parameter	Type	Values	Comment	Default	Opt
GUI	PTR		A valid GUI PTR		
\$ADDRESS	symb		where put the Reply		
PATH	symb		Initial Path	NULL	YES
NEWFOLDER	symb	£FALSE;£TRUE	Show New Folder Button	FALSE	YES
DESCRIPTION	symb		Description		YES
Return					
RES	NV				
OnError					
RES	symb	ERR			
Example					

15 LIS

LIS is a structure that allows you to save an ordered data vector.

The list can be created from data or empty, to be populated later. It can also be created by parsing a text or as the result of a function such as KB1.

Data can be entered in any position and in the same way can be taken.

The list can also be sorted and cloned. And each element can be associated with an attribute.



15.1 LIS_NEW

LIS_NEW		fnc
Description	Create a LIS empty or with given par	
Related	LIS_CLN	
Remark		

1.LIS_NEW					
Parameter	Type	Values	Comment	Default	Opt
PAR	symp		symp to put in the LIS		YES
Return					
RES	PTR		Pointer to the LIS (detete with TRASH)		
OnError					
Example					
a=LIS_NEW(); >>> a is a empty list PTR					
b=LIS_NEW("1","2","3"); >>> b is list ("1","2","3") PTR					

15.2 LIS_CLN

LIS_CLN		fnc
Description	Create a new LIS clone of another LIS	
Related	LIS_NEW	
Remark		

1.LIS_CLN					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		PTR to the LIS to clone		
Return					
RES	PTR		Pointer to the LIS (detete with TRASH)		
OnError					
ERR	symb	NULL			
Example					
<pre>a=LIS_NEW("1","2","3"); b=LIS_CLN(a); >>> b is the copy of a</pre>					



15.3 LIS_SMF

LIS_SMF		fnc
Description	operation on LIS semaphore	
Related	DOOR, SMF	
Remark	deprecated use the SMF of the PTR	

1.LIS_SMF		Command on a SMF			
Parameter	Type	Values	Comment	Default	Opt
PTR to LIS	PTR		PTR to LIS		
cmd	symb	£LCKS(lock 1 level); £LCKR (unlock 1 level); £RST (fully unlock); £TRY(try to lock); £OWN(return the thread owning); £WAIT(wait until is Locked) ; £WAITCNT(count of waiting Thr)	Possible command		YES
Return					
Locks	NV		Number of locks for the caller thread		
Status	NV		if cmd N.D		
Thread	NV		Thread ID of the owner		
OnError					
Example					

15.4 LIS_NUM

LIS_NUM		fnc
Description	The number of symbol contained in the list	
Related		
Remark		

1.LIS_NUM					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
Return					
RES	NV		number of symbol		
OnError					
Example					
<pre>a=LIS_NEW(); LIS_NUM(a); >>> 0 b=LIS_NEW("1","2","3"); LIS_NUM(b); >>> 3</pre>					



15.5 LIS_USE

LIS_USE		fnc
Description	Use content of a list as parameter for function	
Related		
Remark	the LIS is preserved	

1.LIS_USE					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
SEPARATOR	symb		symb used to separate item		yes
TERMINATOR	symb		symb added to the end		yes
Return					
RES	slist		all the symb of the list are pushed in the stack of the function		
OnError					
Example					
<pre>a=LIS_NEW("1","2","3"); CAT(LIS_USE(a)); >>> "123" CAT(LIS_USE(a,"",";")); >>> "1,2,3;"</pre>					

15.6 LIS_PSH

LIS_PSH		fnc
Description	Push an slist of symbols at the begin of a LIS (stack way)	
Related		
Remark		

1.LIS_PSH					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
PARLIST	slist		symbol to push		
Return					
RES	PTR		A valid PTR to LIS		
OnError					
RES	NV	NULL	Invalid PTR		
Example					
<pre>a=LIS_NEW("1","2","3"); LIS_PSH(a,"a","b","c"); >>> list ("c","b","a","1","2","3")</pre>					



15.7 LIS_POP

LIS_POP		fnc
Description	Get out the first symbol of LIS	
Related		
Remark	After the POP the first remaining symbol became the current	

1.LIS_POP					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
Return					
RES	symb		the first symb of the list		
RES	NV	NULL	List empty		
OnError					
RES	NV	NULL	Invalid PTR		
Example					
<pre>a=LIS_NEW("1","2","3"); LIS_POP(a); >>> 1 LIS_NUM(a); >>> 2 LIS_POP(a); >>> 2 LIS_NUM(a); >>> 1 LIS_POP(a); >>> 3 LIS_NUM(a); >>> 0 LIS_POP(a); >>> NULL LIS_NUM(a); >>> 0</pre>					

15.8 LIS_ADD

LIS_ADD		fnc
Description	Add a list of symbols at the end of a list	
Related		
Remark		

1.LIS_ADD					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
PARLIST	slist		symbols to add		
Return					
RES	PTR		ptr to LIS		
OnError					
RES	NV	NULL			
Example					
<pre>a=LIS_NEW("1","2","3"); LIS_ADD(a,"a","b","c"); >>> list ("1","2","3","a","b","c")</pre>					



15.9 LIS_INS

LIS_INS		fnc
Description	Insert a symbol in the LIS	
Related		
Remark		

1.LIS_INS					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
WHERE	symb	£BEF(before); £AFT(after); £BEG(begin); £END(end)	£BEF and £AFT are relative to the cursor current position		
PARLIST	symb		Symb to insert		
Return					
RES	PTR				
OnError					
RES	NV	NULL			
Example					
<pre>a=LIS_NEW("1","2","3"); LIS_INS(a,2,"a","b","c"); >>> list ("1","a","b","c","2","3") LIS_INS(a,£END,"The","End"); >>> list ("1","a","b","c","2","3","The","End")</pre>					

15.10 LIS_GET

LIS_GET		fnc
Description	Get out the symbol at the current position or if provided , the position number	
Related		
Remark		

1.LIS_GET					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
POS	NV		1bsd position		YES
Return					
RES	symb		the symb at the current position		
OnError					
RES	NULL				
Example					
<pre>a=LIS_NEW("1","2","3"); LIS_GET(a); >>> "3" >>> list ("1","2")</pre>					



15.11 LIS_POS

LIS_POS		fnc
Description	Set the current position in the list and return the symbol	
Related		
Remark		

1.LIS_POS					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
POS	symb	###(NV request symbol position 1bsd); £BEG (begin of list);£END (end of list);£NEXT (next); £PRV (previous); £CUR; £ATT(Attribute of the current symbol)	£NEXT and £PRV are relative to the current position, With £CUR the current position is unchanged		
NEWVAL	symb		Set a new value		yes
OPTION	symb	£ATT	if POS is a NV		
Return					
RES	symb		the symb at the current position		
OnError					
RES	NV	NULL			
Example					

15.12 LIS_CLR

LIS_CLR		fnc
Description	Clear a LIS	
Related		
Remark		

1.LIS_CLR					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
Return					
RES	PTR				
OnError					
RES	symb	NULL			
Example					
<pre>lis=LIS_NEW(1,2,3); LIS_NUM(lis); >>> 3, number of elements LIS_CLR(lis); LIS_NUM(lis); >>> 0</pre>					



15.13 LIS_SORT

LIS_SORT		fnc
Description	Order a lis by Val or Att	
Related		
Remark		

1.LIS_SORT					
Parameter	Type	Values	Comment	Default	Opt
LIS	PTR		A valid PTR to LIS		
ORDER	symb	£ASC;£DSC		£ASC	YES
BY	sym	£VAL;£ATT		£VAL	YES
Return					
RES	PTR				
OnError					
RES	symb	NULL			
Example					
<pre>lis=LIS_NEW(2,1,11,31,3); LIS_SORT(lis); CAT(LISUSE(lis,""));</pre> <div>>>> 1,2,3,11,31</div>					

15.14 LIS_SRCH

LIS_SRCH		fnc
Description	Search a symbol in a list.	
Related		
Remark	The comparison is case insensitive and initial & final space insensitive.	

1.LIS_SRCH					
Parameter	Type	Values	Comment	Default	Opt
PAR	PTR		A valid PTR to LIS		
SEARCH	symp		the symbol to search		
Return					
POS	NV		the position 1bsd		
POS	NV	0	NOT found		
OnError					
RES	symp	NULL			
Example					
<pre>lis=LIS_NEW(&pietro,&paolo,&filippo); LIS_SRCH(lis,&paolo); >>> 2, "paolo" is in the second position LIS_SRCH(lis,&attilio); >>> 0, not found</pre>					



16

INI

16.1 INI_NEW

INI_NEW		fnc
Description	Create an INI object if file name is given that file is opened.	
Related	INI_LOD	
Remark	Default delimiter id semi column also # is used	

1.INI_NEW					
Parameter	Type	Values	Comment	Default	Opt
FILE	symb		File to Load		YES
DELIMITER	symb		Delimiter	;	YES
Return					
INI	PTR		Pointer to INI		
OnError					
Example					
<pre>iniP=INI_NEW(); >>> Create empty ini file iniP=INI_NEW("test.ini"); >>> Open "test.ini" (also if file not exists -> empty)</pre>					



16.2 INI_LOD

INI_LOD		fnc
Description	Load a file	
Related		
Remark		

1.INI_LOD					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
FILE	symb		File to load		
DELIMITER	symb		Delimiter	;	YES
Return					
RES	NV	1	Opened		
OnError					
RES	symb	NULL			
Example					
INI_LOD(iniP,"test.ini") >>> open test.ini in preexisting iniP pointer					

16.3 INI_SAV

INI_SAV		fnc
Description	Save INI file	
Related		
Remark		

1.INI_SAV					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
FILE	symb		File to save		
Return					
RES	NV	1	OK		
OnError					
RES	symb	NULL			
Example					
INI_SAV(iniP)					



16.4 INI_LBLNUM

INI_LBLNUM		fnc
Description	Number of label	
Related		
Remark		

1.INI_LBLNUM					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
Return					
NUM	NV		number of Label		
OnError					
RES	symb	NULL			
Example					
INI_LBLNUM(iniP) >>> number of label (0 for empty ini)					

16.5 INI_KEYNUM

INI_KEYNUM		fnc
Description	Number of key inside a label	
Related		
Remark		

1.INI_KEYNUM					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
LABELnum	NV		Number of label instance 1bsd	IN ALL LABEL instance	YES
Return					
NUM	NV		number of key instance		
OnError					
RES	symb	NULL			
Example					
INI_KEYNUM(iniP,£Server) >>> number of keys inside label £Server					



16.6 INI_LBLCOUNT

INI_LBLCOUNT		fnc
Description	Number of instance of a label	
Related		
Remark		

1.INI_LBLCOUNT					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
Return					
NUM	NV		number of instance		
OnError					
RES	symb	NULL			
Example					
INI_LBLCOUNT(iniP,£Server) >>> Instance of label £Server					

16.7 INI_KEYCOUNT

INI_KEYCOUNT		fnc
Description	Number of instance of a key inside a label	
Related		
Remark		

1.INI_KEYCOUNT					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
KEY	symb		Key name		
LABELnum	NV		Number of label instance 1bsd	IN ALL LABEL instance	YES
Return					
NUM	NV		number of key instance		
OnError					
RES	symb	NULL			
Example					
INI_KEYCOUNT(iniP,£Server,£IP) >>> Instances of key IP in label £Server					



16.8 INI_LBLLIST

INI_LBLLIST		fnc
Description	Name of the label	
Related	INI_LBLNUM	
Remark		

1.INI_LBLLIST					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABELnum	NV		Number of label instance	1	YES
Return					
LABEL	symb		label name		
OnError					
RES	symb	NULL			
Example					
INI_LBLLIST(iniP,1) >>> First label of ini					

16.9 INI_KEYLIST

INI_KEYLIST		fnc
Description	name of a key inside a Label	
Related		
Remark		

1.INI_KEYLIST					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
LABELnum	NV		Number of label instance	IN ALL LABEL instance	YES
KEYnum	NV		Number of key instance	1	YES
Return					
KEY	symb		Key name		
OnError					
RES	symb	NULL			
Example					
INI_KEYLIST(iniP,£Server) >>> First key of first instance of £Server label					



16.10 INI_KEYGETSET

INI_KEYGETSET		fnc
Description	Get the Value of key if not exist set with a value	
Related		
Remark	Operate on the first instance	

1.INI_KEYGETSET					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
KEY	symb		Key Name		
VALUE	symb		Key value		
Return					
KEYVAL	symb		value of the Key		
OnError					
RES	symb	NULL			
Example					
<pre>INI_KEYGETSET(iniP,£Server,£IP,"192.168.2.144") >>> se value of key £IP in £Server label</pre>					

16.11 INI_KEYGET

INI_KEYGET		fnc
Description	Get the Value of key	
Related		
Remark		

1.INI_KEYGET					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
KEY	symb		Key Name		
LABELnum	NV		Number of label instance	IN ALL LABEL instance	YES
KEYnum	NV		Number of key instance	1	YES
Return					
KEYVAL	symb		value of the Key		
OnError					
RES	symb	NULL			
Example					
INI_KEYGET(iniP.fServer,fIP) >>> "192.168.2.144"					



16.12 INI_KEYSET

INI_KEYSET		fnc
Description	Set the value of a key	
Related		
Remark		

1.INI_KEYSET					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
KEY	symb		Key Name		
KEYVALUE	symb		Key value		
LABELnum	NV		Number of label instance	1	YES
KEYnum	NV		Number of key instance	1	YES
Return					
RES	nv	1	OK		
OnError					
RES	symb	NULL			
Example					
INI_KEYSET(iniP,£Server,£IP,"192.168.2.144") >>> se value of key £IP in £Server label					

16.13 INI_LBLADD

INI_LBLADD		fnc
Description	Add a label if not already exist if ALWAYS is set ad a new one	
Related		
Remark		

1.INI_LBLADD					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
ALWAYS	symb	NULL;£TRUE	Add also if exists	£TRUE	YES
Return					
RES	NV	1	OK		
OnError					
RES	symb	NULL			
Example					
INI_LBLADD(iniP,£Server) >>> add label £Server to iniP					



16.14 INI_LBLDEL

INI_LBLDEL		fnc
Description	Delete a label instance	
Related		
Remark		

1.INI_LBLDEL					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
LABELnum	NV		Number of label instance	ALL LABEL instance	YES
Return					
RES	NV	1	OK		
OnError					
RES	symb	NULL			
Example					
INI_LBLDEL(iniP,£Server) >>> delete label £Server from iniP					

16.15 INI_KEYADD

INI_KEYADD		fnc
Description	Add a Key to a label	
Related		
Remark		

1.INI_KEYADD					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
KEY	symb		Key Name		
KEYVALUE	symb		Key value		
LABELnum	NV		Number of label instance	1	YES
Return					
RES	NV	1	OK		
OnError					
RES	symb	NULL			
Example					
<pre>INI_KEYADD(iniP,£Server,£PORT,5000) >>> add key £PORT to label £Server with value 5000</pre>					



16.16 INI_KEYDEL

INI_KEYDEL		fnc
Description	Delete a Key instance	
Related		
Remark		

1.INI_KEYDEL					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
LABEL	symb		Label name		
KEY	symb		Key Name		
LABELnum	NV		Number of label instance	ALL LABEL instance	YES
KEYnum	NV		Number of key instance	ALL LABEL instance	YES
Return					
RES	NV	1	OK		
OnError					
RES	symb	NULL			
Example					
INI_KEYDEL(iniP,£Server,£PORT) >>> delete key £PORT from label £Server					

16.17 INI_CLEAR

INI_CLEAR		fnc
Description	Clear INI	
Related		
Remark		

1.INI_CLEAR					
Parameter	Type	Values	Comment	Default	Opt
INI	PTR		Pointer to INI		
Return					
RES	NV	1	OK		
OnError					
RES	symb	NULL			
Example					
INI_CLEAR(iniP) >>> flush iniP pointer					



17

NTW

17.1 NTW_GHBN

NTW_GHBN		fnc
Description	Get Host by name (internet name resolution)	
Related		
Remark		

1.NTW_GHBN					
Parameter	Type	Values	Comment	Default	Opt
NAME	symb		Host name		
Return					
RES	symb		IA of host		
OnError					
RES	symb	NULL			
Example					
NTW_GHBN("WIN-FPHJIHOAIPM") >>> "172.31.63.234"					



17.2 NTW_GHBA

NTW_GHBA		fnc
Description	Get Host by Address (internet name resolution)	
Related		
Remark		

1.NTW_GHBA					
Parameter	Type	Values	Comment	Default	Opt
NAME	symb		Host name		
Return					
RES	symb		Host Name		
OnError					
RES	symb	NULL			
Example					
NTW_GHBA("192.168.2.144") >>> "WIN-FPHJIHOAIPM"					

17.3 NTW_PING

NTW_PING		fnc
Description	Ping Host	
Related		
Remark		

1.NTW_PING					
Parameter	Type	Values	Comment	Default	Opt
HOST	symb		Host		
Return					
RES	NV	1	success		
RES	NV	0	not ping		
OnError					
RES	symb	NULL			
Example					
NTW_PING("192.168.2.144") >>> 1, pc is responding					
NTW_PING("192.168.2.145") >>> 0, pc is not responding					



18

PKG

18.1 PKG_OPN

PKG_OPN		fnc
Description	Open a Package without arguments open the current assembly package	
Related		
Remark		

1.PKG_OPN					
Parameter	Type	Values	Comment	Default	Opt
NAME	symb		Package name		YES
NAME	symb	NULL;£NEW	if not exists create a new pkg		YES
Return					
RES	PTR		PTR to package		
OnError					
RES	symb	NULL			
Example					
PKG_OPN("c:\PWK-PRG\stressGui.pwk") >>> pointer to PKG					



18.2 PKG_MNFGET

PKG_MNFGET		fnc
Description	Giving a Section and a name of a package item return the value	
Related		
Remark	If PKG is null current assembly package is assumed	

1.PKG_MNFGET					
Parameter	Type	Values	Comment	Default	Opt
PKG	PTR		PTR		YES
SECTION	symb		Package Section name		
NAME	symb		Item name		
Return					
RES	symb		Item value		
OnError					
RES	symb	NULL			
Example					
PKG_MNFGET(pkgPtr,£Resources,£BKG) >>> "SFD_YELLOW_1280x1024.png"					

18.3 PKG_MNFADD

PKG_MNFADD		fnc
Description	Add or change an item value in a section	
Related		
Remark	If PKG is null current assembly package is assumed	

1.PKG_MNFADD					
Parameter	Type	Values	Comment	Default	Opt
PKG	PTR		PTR		YES
SECTION	symb		Package Section name (must exist)		
NAME	symb		Item name		
VALUE	symb		item value		
Return					
RES	symb		Item value		
OnError					
RES	symb	NULL			
Example					
PKG_MNFADD(pkgPtr,£Resources,£BKG,"newBackgroud.png") >>> Change background image					



18.4 PKG_SAV

PKG_SAV		fnc
Description	Save back the package	
Related		
Remark	If PKG is null current assembly package is assumed; Save only items declared into the package. This function does NOT WORK in WorkBench. If nor SECTION and ELEMENT are defined all package is saved back. If only SECTION is defined the all SECTION contents are saved. If PKG is null assembly package is assumed.	

1.PKG_SAV					
Parameter	Type	Values	Comment	Default	Opt
PKG	PTR		PTR		YES
SECTION	symb				YES
ELEMENT	symb				YES
Return					
OnError					
Example					
PKG_SAV(pkgPtr,£Resources,£BKG) >>> Save Resources.BKG of the package					

18.5 PKG_PTH

PKG_PTH		fnc
Description	Get the package path	
Related		
Remark	If PKG is null assembly package is assumed	

1.PKG_PTH					
Parameter	Type	Values	Comment	Default	Opt
PKG	PTR		PTR		YES
Return					
RES	symb		Package path		
OnError					
RES	symb	NULL			
Example					
PKG_PTH(pkgPtr) >>> "C:\PWK-TMP\stressGui-1590739115-2416\ "					
PKG_PTH() >>> "C:\PWK-TMP\PWK-ISP-07-1590671330-13252\ " current package					



19

KB1

Knowledge Base 1 (KB1) is an All-In-Memory powerful tools for handling middle amount of data in three structures.

It is constituted by a forest of Items each of them, like tree, can have child Items.

Items are identified by a name unique for the level in which are declared, and can have :

- a value;
- TAG, multiple tags are allowed;
- ATTRIBUTE: a pair attribute-value, ATTRIBUTE can have TAGs .

For ITEM name you should avoid to use number, if you do, to reach it prepend \ (e.g. \88) or use REF() function (e.g. REF("88")).

The main way to interact with KB1 is the KB1_DLG function that accept text statements and can return results as plain TEXT, LIS of results or LIS of TREE items, TBL.

With KB1_DLG you can create, delete, search, compare ITEMS, ATTRIBUTE, TAGs in many ways.

Multi statements can be executed with a single KB1_DLG, each return its result, operators are provided to enable or clear the return of this results: CLR clear all previous results, CLRr clear the results returned after it, since a CLRr.

In each KB1 an item \THISKB is created and contains the PWK_PTR of the KB, to allow the call PWK KB1 function from KB1.

19.1

PWK function

Starting from Ver. 5, PWK function can be invoked by a kb1 statement (dialog).

Attention should be posed to PTR:

inside KB1, LIS and TBL are managed in a different format. When an external function (PWK) is invoked with such PTR, the wrapper (KB1 ↔ PWK) convert this that PTR as a standard PWK PTR, when the function return it check the result and if this PTR is not found it is deleted.

As consequence:

- you can NOT store/use this PTR (e.g in a TBL item) outside the function;
- if the function return the PTR you should delete it.

19.2 KB1 syntax and function

ITEMS if they are root, item are referenced by their name alone or with \ (backslash) prepended (e.g. \alfa), if they are sub item the full path to them must be used.

Tags are label without a value that are tested for existence; the symbol for Tag reference is the double dot (e.g. \alfa:letter).

Attributes are indicated with a dot before the attribute name (e.g. \alfa.weight), attribute have value and can have TAGS that are referenced with a double dot after the single dot (e.g. \alfa.weight:kg).

Items and Attribute can be searched for Having or NOT having TAGs; multi tag is an enumeration of tags separated by comma enclosed by ' (apex). The ! (not) symbol prepended to a tag denote exclusion.

NOTE. Also a single Tag with a ! should be passed enclosed by ' (apex),

The Backslash	
\	alone represent the root and used as operator return all the items at root level
\aa	followed by an item name reference a root item
Multi Backslash : if alone the first represent the root (ideal) item	
\\, x\	return all the items of the first sub level
\\\, x\\	return all the items, if instead a starting item is used, it is returned with all the items below.
Note: multi delete is not allowed. You can't \\=@; x\\=@ but you can \=@ x\\=@	

ITEM should be created with assignment:

Item Creation			
Type	Example	Description	
BY VALUE	item=;	void assignment is valid	
	item='name';		
BY TAG	item:=;	void assignment is valid	
	item:='ship';		
	item:='ship,sailing_boat'	Multi tags	
BY ATTRIBUTE	item.color=;	void assignment is valid	
	item.color='red';		
BY ATTRIBUTE TAG	item.red,='color'		

Item operations			
Operation	Example	Description	
UPDATE	item='new name';		
DELETION	item=@;		

Tag operations			
Operation	Example	Description	
CHANGE	item:ship='boat';	ship is changed in boat	
DELETION	item:boat=@;	boat tag is deleted	
	item:=@	all tags are deleted	



Attribute operations			
Operation	Example	Description	
CHANGE	item.color='blue';	value changed	
DELETION	item.color=@;	attribute color is deleted	
	item.=@;	all attribute are deleted	
TAG	item.color:='prop'	Tag assignment	

Basic Queries (BQ)		
Operation	Operators	Example, Description
ITEM	\	return all root item
	\\	return all items below \ (root)
	\\\	return all items starting from \ (root)
	item\	return item below to item (one level)
	item\\	return all items below item
	item	return item value
TAG	:	return all tags declared in KB1
	item:	return all tags of the item
	item:tag	return tag if declared for item
	item:'tag1,tag2'	if declared both return 'tag1,tag2' else NULL
	:tag	return all item having the tag
	:tag1,tag2,!tag3'	return all item having [tag1,tag2] but NOT [tag3]
ATTRIBUTE	,	return all attribute declared in kb1*
	item.	return all attributed for item
	item.attribute	return the value of the attribute
	.attribute	return all items having the attribute
ATTRIBUTE TAG	\.:	return all tags declared in kb1
	item.:	return all TAGs declared for item attribute*
	item.attribute:	return attribute tag
	item.:tag	return all ATTRIBUTE with the tag
	item.:tag1,!tag2'	return all ATTRIBUTE having [tag1] but not [tag2]
	item.attribute:tag	return tag or null

* for that level

KB1 Operators ans functions	
!! \$ £ € { } () [] , ~ = // / + - * == <> >< < > <= >= & % ^ ! @ CT NCT ? ?? << >> <=> ++ .. ref #WHILE #IF #ELSE #END #SKIP #BREAK RSHF LSHF EQ NE LT GT LE GE NOT AND XOR OR ZNOT ZAND ZOR ZXOR ZSUM PAND POR PXOR MIN MAX IF SIN COS TAN SINH COSH TANH ASIN ACOS ATAN SQRT EXP LOG LOG10 MOD ABS CMP COUNTLIS GETREF EXTRACT UNLINK EXISTS ISNUM ISTRUE ISNULL HELP TBLATT FRM EXC TOTXT CLR CLR\$ CLRr IDCHG	

In BQ you can use comparison operators, multiple assignment.	
\>10	return all Items having value gt 10
\=@	delete all items
.color='green'	Set the value of color to green for all items having color attribute
\:bici or .altezza	return items having BICI tag OR altezza attribute
\:bici and .altezza	return items having BICI tag BOTH altezza attribute
(\\\.Altezza > 20)='MAX'	.Altezza is set to MAX for all items having it >20



19.3 By Example

Create, set, change						
Dialog	Item			Attribute		
	ID	Value	TAG	ID	Value	TAG
i1=	i1					
i2='pippo'	i2	pippo				
i3.height=	i3			height		
i4.height=170	i4			height	170	
i5:='person'	i5		person			
i6.kg:='weight'	i6			kg		weight
i5:person='human'	i5		human			
i6.kg=@	i6					
i6.weight:='kg'	i6			weight		kg
i3.height=50	i3			height	50	
i3.height:='mm'	i3			height	50	mm
i1\h1=	h1					
i2\r1=	r1					
i2\r1\u2=	u2					

Queries (TEXT result)		
Dialog	Result	Comment
\	i1 i2 i3 i4 i5 i6	All item Fst level
\\	h1 r1	Item 2 nd level
\\\	i1 i2 i3 i4 i5 i6 i1\h1 i2\r1 i2\r1\u2	All Items
i2\	r1	
i2\\	i2\r1 i2\r1\u2	
\.	height weight	
\.height	i3 i4	
\.:	mm kg	
\.:mm	\i3.height	
i3.height	50	
i3.:mm	height	
\.:mm==50	\i3.height	
\.height >= 170	i4	

19.4 Knowledge Base 1 - internal function

Special KB1 Operators and Functions			
Op/FN	Symbolic	Description	Note
OP	?	convert an XLIS to SLIS	
OP	??	Convert an XLIS to a SLIS of element with path	
OP	€	External reference of a symbol of the main program	
OP	CT	Contains : enumeration search	
OP	NCT	NOT Contains : enumeration search	
OP	CLR	Clear Results	
OP	CLR _s	Enable Clear result flag	
OP	CLR _r	Disable Clear result flag	
OP	<>	Range External	Not supported
OP	><	Range Internal	Not supported
FN	COUNTLIS	LIS count	
FN	GETREF	given an item name extract an XLIS to use with KB1_XCOPY	
FN	EXTRACT	extract an XLIS to use with KB1_XNS	Get a copy
FN	UNLINK	extract an XLIS to use with KB1_XNS	Remove
FN	EXISTS	In the Item exists 1 else 0	
FN	TBLATT	From an XLIS or Slist return a TBL on an enumeration of ATTRIBUTE and TAG	
FN	FRM	£INS or £DEL a Formula	
FN	EXC	Execute a formula	
FN	TOTXT	giving a KB1 result (XList or Slist or XTbl) return a symb of concat item	def separator ;
FN	IDCHG	To change the Item ID	



19.4.1 COUNTLIS

COUNTLIS		KB1-fnc
Description	Return the number of element of a SLIS or XLIS returned by query	
Related		
Remark	The S/XLis is deleted	

1.COUNTLIS					
Parameter	Type	Values	Comment	Default	Opt
LIS	PTR				
Return					
RES	NV		Number of elements		
OnError					
ERR	symb	NULL			
Example					

19.4.2 GETREF

GETREF		KB1-fnc
Description	Return an XLIS of ITEMS useful to be copied with KB1_XCOPY	
Related		
Remark		

1.GETREF					
Parameter	Type	Values	Comment	Default	Opt
QUERYres	PTR		SLIS or XLIS		
Return					
RES	PTR		XLIS		
OnError					
ERR	symb	NULL			
Example					

19.4.3 EXTRACT

EXTRACT		KB1-fnc
Description	Return an XLIS of COPIED ITEMS useful to use with KB1_XNS	
Related	UNLINK	
Remark	The function accept multi parameters of both type (Queries or with name change)	

1.EXTRACT					
Parameter	Type	Values	Comment	Default	Opt
QUERYres	PTR		XLIS or SLIS		YES
ItemName=NewName	symb		If =NewName exists the Item get this name in the res XLIS		YES
Return					
RES	PTR		XLIS		
OnError					
ERR	symb	NULL			
Example					
xl=EXTRACT(\MAT\); >> extract all elements below MAT xl=EXTRACT(\ART\Cod1=Pippo'); >> extract Cod1 and rename it as Pippo xl=EXTRACT(\MAT\, \ART\Cod1=Pippo') >>> Both are added to xl					

19.4.4 UNLINK

UNLINK		KB1-fnc
Description	Return an XLIS of REMOVED ITEMS useful to use with KB1_XNS	
Related	EXTRACT	
Remark	The function accept multi parameters of both type (Queries or with name change)	

1.UNLINK					
Parameter	Type	Values	Comment	Default	Opt
QUERYres	PTR		XLIS or SLIS		YES
ItemName=NewName	symb		If =NewName exists the Item get this name in the res XLIS		YES
Return					
RES	PTR		XLIS		
OnError					
ERR	symb	NULL			
Example					
xl=EXTRACT(\MAT\); >> extract all elements below MAT xl=EXTRACT(\ART\Cod1=Pippo'); >> extract Cod1 and rename it as Pippo xl=EXTRACT(\MAT\, \ART\Cod1=Pippo') >>> Both are added to xl					

19.4.5



19.4.6 TBLATT

TBLATT		KB1-fnc
Description	Return a PTR to TBL from the result of a query	
Related		
Remark		

1.TBLATT					
Parameter	Type	Values	Comment	Default	Opt
QRYRES	PTR				
ATTRIBUTE	enum				YES
SEP	synb		separator of the attribute enum	;	YES
TBLatt	symb	£NO;£YES	TBL	£NO	YES
TAGenum	symb	£NO;£YES	TBL	£NO	YES
SubQuery_1	symb				YES
SubQuery_x	symb				YES
Return					
RES	PTR		To TBL		
OnError					
ERR	symb	NULL			
Example					
<p>In column 1 : _ITEM_ Item name In column 2: _VAL_ Item value</p> <p>If(TBLatt == £YES) In Column 3: _ TBLATT_ PTR to attribute TBL</p> <p>I(TAGenum == £YES) In Column 3 (or 4 if TBLatt) _TAG_ comma separated enum of TAG</p> <p>If ATTRIBUTE enum is defined the following column contains the values of the attribute and the column name is the attribute name.</p> <p>Row name are set with the full ITEM path name</p> <p>In SubQuery (you can add as many queries you want), you can use _PATH_, _ITEM_, _VAL_ that are replaced before its execution, by the values obtained by the Main Query; results are returned in the relative column and you can set the column name adding " AS colname" at the end of the SQ.</p>					

19.4.7 IDCHG

IDCHG		KB1-fnc
Description	Test the existence of an Item and/or of an attribute of an item	
Related		
Remark		

1.IDCHG					
Parameter	Type	Values	Comment	Default	Opt
ITEM_ID	symb		Item ID as £		
NEW_ID	symb		new ID as £		
Return					
RES	NV		new ID		
OnError					
ERR	symb	NULL			
Example					

19.4.8 EXISTS

EXISTS		KB1-fnc
Description	Test the existence of an Item and/or of an attribute of an item	
Related		
Remark		

1.EXISTS					
Parameter	Type	Values	Comment	Default	Opt
ITEM(.ATTR)	symb		Item or Item.Attribute		YES
Return					
RES	NV		1=exists , 0=not exists		
OnError					
ERR	symb	NULL			
Example					



19.5 KB1_OPN

KB1_OPN		fnc
Description	Open a knowledge Base	
Related		
Remark	The FILENAME if defined is used in KB1_SAV as default	

1.KB1_OPN					
Parameter	Type	Values	Comment	Default	Opt
FILENAME	symb		Name of the file containing the KB		YES
TYPE	symb	NULL;£TEXT	Type of the following elem	£TEXT	YES
ELM	symb		if(TYP=TEXT) Element to load instead of the file.		YES
Return					
RES	PTR		Pointer to KB		
OnError					
Example					

19.6 KB1_SMF

KB1_SMF		fnc
Description	operation on KB1 semaphore	
Related	DOOR, SMF	
Remark	deprecated use the smf of the PTR instead	

1.KB1_SMF		Command on a SMF			
Parameter	Type	Values	Comment	Default	Opt
PTR to LIS	PTR		PTR to KB1		
cmd	symb	£LCKS(lock 1 level); £LCKR (unlock 1 level); £RST (fully unlock); £TRY(try to lock); £OWN(return the thread owning)	Possible command		YES
Return					
Locks	NV		Number of locks for the caller thread		
Status	NV		if cmd N.D		
Thread	NV		Thread ID of the owner		
OnError					
Example					



19.7 KB1_SAV

KB1_SAV		fnc
Description	Open a knowledge Base	
Related		
Remark	If FILENAME if N.D. the default is used	

1.KB1_SAV					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR		Pointer to KB		
FILENAME	symb		Name of the file containing the KB		YES
OPTION	symb	£PROTECT; £VISIBLE; £IFCHG; £ASYNC	with £PROTECT KB is kripted	£PROTECT, £SYNC	YES
Return					
RES	PTR		Pointer to KB		
OnError					
Example					
<p>£PROTECT / £VISIBLE are alternative; if you add £IFCHG the save operation is done only if the KB1 is changed since the last save and the name is the default name.</p> <pre>kb1_opn("my.kb1");</pre> <p>kb1_sav(NULL,£VISIBLE) >>> saved with the default name</p> <p>kb1_save(NULL, "VISIBLE IFCHG") >>> save only if changed since the last save</p> <p>kb1_save("new.kb1", "VISIBLE IFCHG") >>> the name is changed so is saved also if not changed</p> <p>If £ASYNC the save operation is made by another thread.</p>					

19.8 KB1_GET

KB1_GET		fnc
Description	Return in a symbol the serialization of the kb	
Related		
Remark		

1.KB1_GET					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR				
TAB	symb	£NOT;£YES	serialization with indent tab ?	£NOT	YES
Return					
RES	PTR		Pointer to KB		
OnError					
Example					



19.9 KB1_DLГ

KB1_DLГ		fnc
Description	Make a query to a knowledge Base	
Related	KB1_QRY, KB1_EXC	
Remark	Lock for Read and Write. See general discussion on the manual PWK-MAN-LIB-CORE	

1.KB1_DLГ					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR		Pointer to KB		
DLGTEXT	symb				
FORMAT	symb	NULL;£TEXT	The result format		YES
SEP	symb		The separator to use for Lis result	<SPACE>	YES
SEPres	symb		separator between results	<CRLF>	YES
Return					
RES	symb		the result		
OnError					
Example					
<p>ITEM:TYPE.ATTRIB</p> <p>ITEM=itemValue >>> itemValue is set ITEM:= typeValue >>> typeValue is add ITEM:tv= typeValue >>> tv is changed with typeValue ITEM.ATTRIB=attribValue</p> <p>\ >>> LIS of item ITEM\ >>> LIS of sub Item ITEM >>> itemValue ITEM: >>> LIS of typeValue ITEM.ATTRIB >>> attribValue ITEM. >>> LIS of ATTRIB</p> <p>To separate statement use the semi colon ; if the Fst() of a result is: 18 a LIS of TREE item is returned and should be destroyed after the use 29 a LIS of SYMB is returned and should be destroyed after the use</p>					

19.10 KB1_QRY

KB1_QRY		fnc
Description	Make a query to a knowledge Base	
Related	KB1_DLG, KB1_EXC	
Remark	Lock for read. See general discussion on the manual PWK-MAN-LIB-CORE,	

1.KB1_QRY					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR		Pointer to KB		
DLGTEXT	symb				
FORMAT	symb	NULL;£TEXT	The result format		YES
SEP	symb		The separator to use for Lis result	<SPACE>	YES
SEPres	symb		separator between results	<CRLF>	YES
Return					
RES	symb		the result		
OnError					
Example					
<p>ITEM:TYPE.ATTRIB</p> <p>ITEM=itemValue >>> itemValue is set ITEM:= typeValue >>> typeValue is add ITEM:tv= typeValue >>> tv is changed with typeValue ITEM.ATTRIB=attribValue</p> <p>\ >>> LIS of item ITEM\ >>> LIS of sub Item ITEM >>> itemValue ITEM: >>> LIS of typeValue ITEM.ATTRIB >>> attribValue ITEM. >>> LIS of ATTRIB</p> <p>To separate statement use the semi colon ; if the Fst() of a result is:</p> <p>18 a LIS of TREE item is returned and should be destroyed after the use 29 a LIS of SYMB is returned and should be destroyed after the use</p>					

**19.11 KB1_EXC**

KB1_EXC		fnc
Description	Make a query to a knowledge Base using an existing Formula	
Related	KB1_DLG, KB1_QRY, KB1_FRM	
Remark	the formula name is normalized : upper case,no spaces	

1.KB1_EXC					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR		Pointer to KB		
FRMname	symb				
FORMAT	symb	NULL;£TEXT	The result format		YES
SEP	symb		The separator to use for Lis result	<SPACE>	YES
SEPres	symb		separator between results	<CRLF>	YES
Return					
RES	symb		the result		
OnError					
Example					

19.12 KB1_FRM

KB1_FRM		fnc
Description	Operation on formulas	
Related	KB1_EXC	
Remark	the formula name is normalized : upper case, no spaces	

1.KB1_FRM					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR		Pointer to KB		
NAME	symb		Name of the formula		
CMD	symb	£INFO(actual formula); £INS(insert or modify); £DEL(delete)		£INFO	YES
FORMULA	symb				YES
Return					
RES	symb		the result depend from CMD		
OnError					
ERR	symb	NULL			
Example					



19.13 KB1_XNS

KB1_XNS		fnc
Description	Insert a XLIS in a knowledge base	
Related	KB_XCP	
Remark	Usually the element of the XLIS are EXTRACT from a KB1	

1.KB1_XNS					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR		Pointer to KB		
XLIS	symb/PTR		"XLIS:12.." or a PTR		
PATH	symb		Where to add		OPT
Return					
XLIS	PTR		PTR to xlis		
OnError					
ERR	symb				
Example					
kb : PTR to KB1 a : could be "XLIS:123456" or a PTR to a LIS containing an XLIS KB1_XNS(kb, a, "pippo");					

19.14 KB1_XCP

KB1_XCP		fnc
Description	Insert a XLIS in a knowledge base, each element is copied from a reference	
Related	KB1_XNS	
Remark	Usually the element of the XLIS are GETREF from a KB1	

1.KB1_XCP					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR		Pointer to KB		
XLIS	symb/PTR		"XLIS:12.." or a PTR		
PATH	symb		Where to add		OPT
Return					
XLIS	PTR		PTR to xlis		
OnError					
ERR	symb				
Example					
kb : PTR to KB1 a : could be "XLIS:123456" or a PTR to a LIS containing an XLIS than contains reference to item KB1_XCP(kb, a, "\pippo");					



19.15 KB1_INF

KB1_INF		fnc
Description	Get KB1 info	
Related		
Remark		

1.KB1_INF					
Parameter	Type	Values	Comment	Default	Opt
KB	PTR				
SEL	symb	£OPFNCLIS; \$CHGCNT			
Return					
RES	PTR		If SEL=OPFNCLIS a PTR to a LIS containing KB1 operators and functions. If SEL=CHGCNT the change counter		
OnError					
Example					

20 MAIL

The following functions are not yet documented: MAIL_SMTP



21

TBL

TBL is a two-dimensional data structure, with columns and rows, equipped with a whole series of functionalities to manage information dynamically.

TBLs can be created:

- 1) simplifying indicating the number of rows and columns
- 2) parserizing a text and indicating the separators of rows and columns
- 3) specifying the names of the columns and the number of rows
- 4) as a result of query DB, KB1 ...

The TBLs can be modified:

- 1) baptizing the names of columns and / or rows
- 2) adding columns and / or rows
- 3) exchanging columns and / or rows
- 4) rearranging the elements
- 5) by cloning them
- 6) by exporting its values

All the functions related to the TBLs are described below.

21.1 TBL_NEW

TBL_NEW		fnc
Description	Create a table from row/column or from a text	
Related		
Remark		

1.TBL_NEW		From row and column			
Parameter	Type	Values	Comment	Default	Opt
NCOL	NV		Number of column		
NROW	NV		Number of rows		
Return					
RES	PTR		Table PTR		
OnError					
RES	symb	NULL			
Example					
TBL_NEW(2,2) >>> Table ("" , "") ("" , "")					

2.TBL_NEW		From row and column names			
Parameter	Type	Values	Comment	Default	Opt
NCOL	NV		Number of column or NULL		
NROW	NV		Number of rows		
ItemSep	symb		Separator between Names	;	YES
GroupSep	symb		IGNORED		YES
COLUMNNames	enum		COLUMN names		YES
Return					
RES	PTR		Table PTR		
OnError					
RES	symb	NULL			
Example					
If NCOL is zero or NULL is inferred by the number of COLUMNNames, If defined it set the Number of Columns					
TBL_NEW(NULL,2,";",NULL,"ID;DSC") >>> Table ("" , "") ("" , "") with column 1 name "ID" and 2 "DSC"					



3.TBL_NEW		From Text			
Parameter	Type	Values	Comment	Default	Opt
TEXT	symb		data of the table		
BY	symb	£COL;£ROW	data are by col or by row		
ItemSep	symb		Separator between item		YES
GroupSep	symb		Separator between group		YES
Return					
RES	PTR		Table PTR		
OnError					
RES	symb	NULL			
Example					
<pre>txt="luca;1;192 pino;2;174"; tbl=TBL_ITM(txt,£ROW,";", " ") >>> Table ("luca" , "1" , "192") ("pino" , "2" , "174") >>> column are "1", "2" and "3" , row are "1" and "2"</pre>					

21.2 TBL_SMF

TBL_SMF		fnc
Description	operation on TBL semaphore	
Related	DOOR, SMF	
Remark		

1.TBL_SMF		Command on a SMF			
Parameter	Type	Values	Comment	Default	Opt
PTR to LIS	PTR		PTR to TBL		
cmd	symb	£LCKS(lock 1 level); £LCKR(unlock 1 level); £RST(fully unlock); £TRY(try to lock); £OWN(return the thread owning); £LCKDSB(Disable TBL semaphore); £LCKENB(Enable standard smf behaviour); £WAIT(wait until is Locked) ; £WAITCNT(count of waiting Thr)	Possible command		YES
Return					
Locks	NV		Number of locks for the caller thread		
Status	NV		if cmd N.D		
Thread	NV		Thread ID of the owner		
OnError					
Example					
£LCKDSB should be used carefully because synchronisation is suppressed, as counter part in this way more then one thread can access the TBL without interlock.					



21.3 TBL_CLN

TBL_CLN		fnc
Description	Clone the TBL in a new TBL	
Related		
Remark		

1.TBL_CLN		Command on a SMF			
Parameter	Type	Values	Comment	Default	Opt
PTR to TBL	PTR		PTR to LIS		
Return					
CLN_TBL	PTR		the PTR of the cloned one		
OnError					
Example					

21.4 TBL_SAV

TBL_SAV		fnc
Description	Save the TBL content in file	
Related	TBL_LOD	
Remark		

1.TBL_SAV		Command on a SMF			
Parameter	Type	Values	Comment	Default	Opt
PTR to TBL	PTR		PTR to LIS		
FILENAME	symb		File Name		
Return					
RES	PTR		The PTR of the TBL		
OnError					
RES	symb	NULL			
Example					
<pre>TBL_SAV(tblPtr,"today.tbl") >> save the content of tblPtr (data, column names and row names) to today.tbl</pre>					



21.5 TBL_LOD

TBL_LOD		fnc
Description	Load the TBL with the content of a file created with a previous TBL_SAV	
Related	TBL_SAV	
Remark		

1.TBL_LOD		Command on a SMF			
Parameter	Type	Values	Comment	Default	Opt
PTR to TBL	PTR		PTR to LIS		
FILENAME	symb		File Name		
Return					
RES	PTR		The PTR of the TBL		
OnError					
Example					
<pre>TBL_LOD(tblPtr,"today.tbl") >>> load the table saved in today.tbl in the tblPtr pointer (overwriting preexistent datas) tblPtr=TBL_LOD(NULL,"today.tbl") >>> load tje table saved in today.tbl in a new table pointer</pre>					

21.6 TBL_CHG

TBL_CHG		fnc
Description	Change the table dimension	
Related		
Remark	If possible the content is preserved; Operator ADD SUB INS DEL can be prep-ended	

1.TBL_CHG					
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
NCOL	symb	£NULL;£ADD; £SUB;£INS; £DEL	new Number of column or unchanged if N.D. INS and DEL alone cause the insertion or the deletion at the First position.	actual number	YES
NROW	symb	£ADD;£SUB; £INS;£DEL	new Number of rows or unchanged if N.D.	actual number	YES
RETVAL	symb	£PTR;£ROW; £COL	The return val	£PTR	YES
Return					
RES	PTR		PTR to the table(default) or row or col number (RETVAL)		
OnError					
RES	symb	NULL	Table PTR		
Example					
TBL_CHG(tbl, 10, 20); !! set to 10 columns and 20 rows;					
TBL_CHG(tbl, null, "ADD 2"); !! add 2 rows :10 columns and 22 rows;					
TBL_CHG(tbl, null, "DEL 2"); !! Row N°2 is deleted;					
TBL_CHG(tbl, "INS 3"); !! A column is inserted BEFORE actual column 3;					
TBL_CHG(tbl, "INS ", "DEL"); !! A column is inserted in the First position and the first row is deleted;					



21.7 TBL_ITM

TBL_ITM		fnc
Description	Get or Set The value of an item	
Related	TBL_NAM	
Remark	If col or row have a name that can be used instead of the number	

1.TBL_ITM		Single insertion / extraction			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
COL	symb		column number or name		
ROW	symb		row number or name		
VAL	symb		if SET the value		YES
Return					
RES	symb		The value of the item		
OnError					
RES	symb	NULL			
Example					

2.TBL_ITM		Multi insertion			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
COL	symb		column number or name		
ROW	symb		row number or name		
SELECTOR	symb	£ROW;£COL	£ROW, £COL	£ROW	
ITEMS	slis				
Return					
RES	PTR		The TBL PTR		
OnError					
RES	symb	NULL			
Example					
<p>In this form can be used for multiple insert. NULL values are skipped. The selector can be followed by a number, in this case a BLOCK insertion is intended and the number indicate the dimension:</p> <p>if SELECTOR == £ROW the number of elements for each row if SELECTOR == £COL the number of elements of each column</p> <p>TBL_ITM(tbl, 1,1, £ROW, a,b,c,d) // row insertion</p> <p>==> ROW 1: a, b, c, d;</p> <p>TBL_ITM(tbl, 2,2, "ROW 2", a,b,c,d) // row insertion (x= existing element)</p> <p>==> ROW 2: x, a, b; ==> ROW 3: x, c, d;</p>					

21.8 TBL_NAM

TBL_NAM		fnc
Description	Assign names to columns or rows	
Related	TBL_ITM	
Remark		

1.TBL_NAM		Single Assignment			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
WHAT	symb	£COL;£ROW			
NUM	NV		Number 1bsd of the col or row		
NAME	symb		The name		
Return					
RES	PTR		Table PTR		
OnError					
RES	symb	NULL			
Example					
TBL_NAM(tblPtr,£COL,5,£SEL) >> name the column 5 of the tblPtr as "SEL". You can refer it as 5 or "SEL"					

2.TBL_NAM		Multi assignment			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
WHAT	symb	£COL;£ROW			
TEXT	enum		Enumeration of names		
SEP	symb		Names separator		
Return					
RES	PTR		Table PTR		
OnError					
RES	symb	NULL			
Example					
<pre>TBL_NAM(tblPtr,£COL,"ID;DSC;SEL",";") >>> name the columns of tblPtr : 1=ID, 2=DSC, 3=SEL TBL_NAM(tblPtr,£COL,2) >>> £DSC</pre>					



21.9 TBL_INF

TBL_INF		fnc
Description	TBL info	
Related		
Remark		

1.TBL_INF		Without PAR			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
WHAT	symb	£COL;£ROW			
Return					
RES	NV		Number of cols if WHAT=£COL		
RES	NV		Number of rows if WHAT=£ROW		
OnError					
RES	symb	NULL			
Example					
tblPtr=TBL_NEW(NULL,2,";",NULL,"ID;DSC;SEL");					
TBL_INF(tblPtr,£COL) >>> 3					
TBL_INF(tblPtr,£ROW) >>> 2					

2.TBL_INF		With PAR			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
WHAT	symb	£COL;£ROW			
PAR	symb		Name or Number (1bsd)		
Return					
RES	NV		if PAR is a col or row Name		
RES	symb		if PAR is a col or row Number		
OnError					
RES	symb	NULL			
Example					
tblPtr=TBL_NEW(NULL,2,";",NULL,"ID;DSC;SEL");					
TBL_INF(tblPtr,£COL,3) >>> "SEL"					

3.TBL_INF		enum of cols or rows names			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
WHAT	symb	£COLNAMES; £ROWNAMES			
SEP	symb		enum separator	;	YES
Return					
RES	symb		enum of names		
OnError					
RES	symb	NULL			
Example					
tblPtr=TBL_NEW(NULL,2,";",NULL,"ID;DSC;SEL");					
TBL_INF(tblPtr,£COLNAMES) >>> "ID;DSC;SEL"					

4.TBL_INF		Name of one row or col			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
WHAT	symb	£COLNAME; £ROWNAME			
INDEX	NV				
Return					
RES	symb		Name		
OnError					
RES	symb	NULL			
Example					



21.10 TBL_EXP

TBL_EXP		fnc
Description	Export the table content all or partial in a symbol	
Related	TBL_IMP TBL_USE	
Remark		

1.TBL_EXP					
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
itemSep	symb		Separator between item	;	YES
groupSep	symb		Separator between group	CRLF	YES
BY	symb	£COL;£ROW		£ROW	YES
STARTcol	NV		From col		YES
STARTrow	NV		From row		YES
COLnum	NV		Number of cols		YES
ROWnum	NV		Number of rows		YES
Return					
RES	symb		data with separator		
OnError					
RES	symb	NULL			
Example					
txt="luca;1;192 pino;2;174"; tbl=TBL_ITM(txt,£ROW,";", " "); TBL_EXP(tbl,";", " ", £COL, 3, 1) >>> "192;174"					

21.11 TBL_USE

TBL_USE		fnc
Description	Export the table content all or partial as separated symbol	
Related	TBL_IMP, TBL_EXP	
Remark		

1.TBL_USE					
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
itemSep	symb		Separator between item		YES
groupSep	symb		Separator between group		YES
BY	symb	£COL;£ROW		£ROW	YES
STARTcol	NV		From col		YES
STARTrow	NV		From row		YES
COLnum	NV		Number of cols		YES
ROWnum	NV		Number of rows		YES
Return					
RES	slis		list of symbol		
OnError					
RES	symb	NULL			
Example					
CAT(TBL_USE(tbl,";", " ", £ROW)) is equivalent to TBL_EXP(tbl,";", " ", £ROW)					

**21.12 TBL_IMP**

TBL_IMP		fnc
Description	Load the table with data	
Related	TBL_EXP TBL_USE	
Remark		

1.TBL_IMP		Rows and columns are adjusted in respect the data			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
DATA	symb		Data		
itemSep	symb		Separator between item	;	YES
groupSep	symb		Separator between group	CRLF	YES
BY	symb	£COL;£ROW		£ROW	YES
Return					
RES	symb		PTR to TBL		
OnError					
RES	symb	NULL			
Example					

2.TBL_IMP		The data are inserted in the existing tbl			
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
DATA	symb		Data		
itemSep	symb		Separator between item	;	YES
groupSep	symb		Separator between group	CRLF	YES
BY	symb	£COL;£ROW		£ROW	YES
STARTCOL	symb		Starting column		
STARTROW	symb		starting row		
Return					
RES	symb		PTR to TBL		
OnError					
RES	symb	NULL			
Example					

21.13 TBL_SRC

TBL_SRC		fnc
Description	Search in col or row for a symb	
Related		
Remark		

1.TBL_SRC					
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
IN	symb	£COL;£ROW			
NUM	symb		number (1bsd) or name of the row or col		
ITEM	symb		symb to search		
START	symb		Start row or col number or name		YES
Return					
RES	NV		Number of the row or col		
OnError					
RES	symb	NULL			
Example					



21.14 TBL_EXG

TBL_EXG		fnc
Description	Exchange the content of two row or col	
Related		
Remark		

1.TBL_EXG					
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
WHAT	symb	£COL;£ROW			
This	symb		number (1bsd) or name of the row or col		
WithThis	symb		number (1bsd) or name of the row or col		
Return					
RES	NV		PTR to TBL		
OnError					
RES	symb	NULL			
Example					
<pre>txt="luca;1;192 pino;2;174"; tbl=TBL_ITM(txt,£ROW,";", " ") TBL_EXG(tbl,£COL,2,3) >>> Table ("luca" , 192 , 1) ("pino" , 174 , 2)</pre>					

21.15 TBL_SORT

TBL_SORT		fnc
Description	Order row or col using a row or col content as key	
Related		
Remark		

1.TBL_SORT					
Parameter	Type	Values	Comment	Default	Opt
TBL	PTR		Valid PTR to a TBL		
WHAT	symb	£COL;£ROW			
KEY	symb		number (1bsd) or name of the row or col		
ORDER	symb	£ASC;£DSC		£ASC	YES
Return					
RES	NV		PTR to TBL		
OnError					
RES	symb	NULL			
Example					
TBL_SORT(tbl,£ROW,£NAME,ASC)					
Note: if the column/row used for sort contains numerical value, the sort considers they as string. So 1, 2, 3, 11, 21 sorted became 1,11,2,21,3					



22

HTTP

22.1 HTTP_OPN

HTTP_OPN		fnc
Description	Create an HTTP connection to a server	
Related		
Remark		

1.HTTP_OPN					
Parameter	Type	Values	Comment	Default	Opt
TYPE	symb	£HTTP;£HTTPS			
HOST	symb		Host name		
PORT	NV		Port		
Return					
RES	NV		PTR to HTTP		
OnError					
RES	symb	NULL			
Example					



22.2 HTTP_GET

HTTP_GET		fnc
Description	Send a GET command to an HTTP server	
Related		
Remark	(MRF) Multi reply Function	

1.HTTP_GET					
Parameter	Type	Values	Comment	Default	Opt
HTTP	PTR		PTR to HTTP		
CMD	symb		HTTP command		
WHAT	symb	£TEXT;£FILE			
ENCODING	symb	£A;£U;£UTF			
FILENAME	symb		WHAT filename		
HEADEROPTION	symb		HTTP Header option		YES
REPLYTYPE	symb	NULL;£HEADER			YES
Return					
RES1	symb		The reply		
RES2	symb		Header of the reply get it with ->		
OnError					
RES1	symb	NULL			
Example					

22.3 HTTP_DEL

HTTP_DEL		fnc
Description	Send a DELETE command to an HTTP server	
Related		
Remark	(MRF) Multi reply Function	

1.HTTP_DEL					
Parameter	Type	Values	Comment	Default	Opt
HTTP	PTR		PTR to HTTP		
CMD	symb		HTTP command		
WHAT	symb	£TEXT;£FILE	for reply		
ENCODING	symb	£A;£U;£UTF	for replay		
FILENAME	symb		WHAT filename		
HEADEROPTION	symb		HTTP Header option		YES
REPLYTYPE	symb	NULL;£HEADER			YES
Return					
RES1	symb		The reply		
RES2	symb		Header of the reply get it with ->		
OnError					
RES1	symb	NULL			
Example					



22.4 HTTP_POST

HTTP_POST		fnc
Description	Send a POST command to an HTTP server	
Related		
Remark	(MRF) Multi reply Function	

1.HTTP_POST					
Parameter	Type	Values	Comment	Default	Opt
HTTP	PTR		PTR to HTTP		
CMD	symb		HTTP command		
WHAT	symb	£TEXT;£FILE			
ENCODING	symb	£A;£U;£UTF			
WHAT-PAYLOAD	symb		filename or TEXT		
RPL-WHAT	symb	£TEXT;£FILE			
RPL-ENCODING	symb	£A;£U;£UTF			
RPL-FILENAME	symb		RPL-WHAT filename		
HEADEROPTION	symb		HTTP Header option		YES
REPLYTYPE	symb	NULL;£HEADER			YES
Return					
RES1	symb		The reply or NULL		
RES2	symb		Header of the reply get it with ->		
OnError					
RES1	symb	-1			
Example					

22.5 HTTP_PUT

HTTP_PUT		fnc
Description	Send a PUT command to an HTTP server	
Related		
Remark	(MRF) Multi reply Function	

1.HTTP_PUT					
Parameter	Type	Values	Comment	Default	Opt
HTTP	PTR		PTR to HTTP		
CMD	symb		HTTP command		
WHAT	symb	£TEXT;£FILE			
ENCODING	symb	£A;£U;£UTF			
WHAT-PAYLOAD	symb		filename or TEXT		
RPL-WHAT	symb	£TEXT;£FILE			
RPL-ENCODING	symb	£A;£U;£UTF			
RPL-FILENAME	symb		RPL-WHAT filename		
HEADEROPTION	symb		HTTP Header option		YES
REPLYTYPE	symb	NULL;£HEADER			YES
Return					
RES1	symb		The reply or NULL		
RES2	symb		Header of the reply get it with ->		
OnError					
RES1	symb	-1			
Example					



23

WUI

23.1 WUI_OPN

WUI_OPN		fnc
Description	Open a WUI	
Related		
Remark		

1.WUI_OPN					
Parameter	Type	Values	Comment	Default	Opt
WUI_ITF	sybm		Name of a .pkw file		YES
Return					
RES	NV	PTR	PTR to WUI		
OnError					
Example					



23.2 WUI_LOD

WUI_LOD		fnc
Description	load a new .pkw file	
Related		
Remark		

1.WUI_LOD					
Parameter	Type	Values	Comment	Default	Opt
WUI	PTR		Valid WUI ptr		
WUI_ITF	symb		Name of a .pkw file		
Return					
RES	NV	PTR	PTR to WUI		
OnError					
ERR	symb	NULL			
Example					

23.3 WUI_SET

WUI_SET		fnc
Description	Depending from the WHT par set the TRIG or the address of the static Web	
Related		
Remark		

1.WUI_SET					
Parameter	Type	Values	Comment	Default	Opt
WUI	PTR		Valid WUI ptr		
WHT	symb	£TRG_HTML; £TRG_SYM; £TRG_INC; £TRG_CTM;£STC		HTML	
PTR	NV		Pointer to TRIG or address of static WEB		
Return					
RES	NV	PTR	PTR to WUI		
OnError					
ERR	symb	NULL			
Example					

**1.WUI_SET**

For INC : "TRG_WUI, TRG_PAGE, TRG_LANG, TRG_XTREE, TRG_XITM, TRG_XTYP, TRG_ID, TRG_ATT,TRG_TYP, TRG_VAL, TRG_PAR, TRG_FROM, TRG_R_XITM, TRG_R_ID, TRG_R_NUM, TRG_KB1, TRG_CTM, TRG_RES, TRG_RES_VAL "

For HTML : "TRG_WUI, TRG_PAGE, TRG_LANG, TRG_XTREE, TRG_XITM, TRG_XTYP, TRG_ID, TRG_ATT, TRG_TYP, TRG_VAL, TRG_PAR, TRG_FROM, TRG_KB1, TRG_CTM, TRG_RES, TRG_RES_VAL "

For SYM : "TRG_WUI, TRG_PAGE, TRG_LANG, TRG_TYP, TRG_VAL, TRG_PAR, TRG_KB1, TRG_CTM, TRG_RES, TRG_RES_VAL "

For CTM : "TRG_WUI, TRG_PAGE, TRG_LANG, TRG_XTREE, TRG_XITM, TRG_XTYP, TRG_ID, TRG_ATT, TRG_TYP, TRG_VAL, TRG_PAR, TRG_FROM, TRG_KB1, TRG_CTM, TRG_RES, TRG_RES_VAL "

In TRG_ID the content of "data-pwk-ctm-id" attribute and in TRG_ATT the attribute name or if TRG_XTYP=TEXT the text content.

TRIG SET parameters

=====

TRG_WUI	PTR	The Wui
TRG_PAGE	symb	The html page name
TRG_LANG	symb	The language ID (from GUI_HTML)
TRG_XTREE	PTR	The Tree of XITM
TRG_XITM	nv	Tree item
TRG_XTYP	symb	TEXT,ATTR,CDTA,ELEM,CMNT
TRG_ID	symb	HTML id or data-pwk-ctm-id (for CTM)
TRG_ATT	symb	if XTYP=ATTR the attribute name
TRG_TYP	symb	£, RES, PKG, REL, SYM, EMB or HTML tag
TRG_VAL	symb	value of Item
TRG_PAR	symb	Parameter set in WUI_HTML
TRG_KB1	PTR	Usually the KB1 of WEB trig (from GUI_HTML)
TRG_FROM	symb	£HTML, £REF, £REF_ADD, £RPT, RPT_ADD
TRG_R_XITM	nv	The replicated REF or RPT elm
TRG_R_ID	symb	The ID of the _REF_ or _RPT_
TRG_R_NUM	nv	The ordinal number of REF RPT replication (times REF parameters)

TRIG GET parameters

=====

TRG_RES	nv	0= translation are not performed, 1=performed
TRG_RES_VAL	symb	Value (if £REF or £REP) is the new value of the times parameter

23.4 WUI_HTML

WUI_HTML		fnc
Description	load a new .pkw file	
Related	WUI_CPTH	
Remark		

1.WUI_HTML					
Parameter	Type	Values	Comment	Default	Opt
WUI	PTR		Valid WUI ptr		
PAGE	symb		ID of the HTML page in wui		
KB1	PTR		Received in TRG_KB1		YES
LANG	symb		Received in TRG_LANG		YES
PAR	symb		For TRG		YES
CTM	PTR		ConTent Manager PTR		YES
Return					
RES	symb		The HTML page ready to be send		
OnError					
ERR	symb	NULL			
Example					



23.5 WUI_CPTH

WUI_CPTH		fnc
Description	Symbolic path translation	
Related	WUI_SPTH	
Remark	WUI_HTML can insert Symbolic Path to item , use this to get the real pth	

1.WUI_CPTH					
Parameter	Type	Values	Comment	Default	Opt
WUI	PTR		Valid WUI ptr		
PTH	symb		Path		
Return					
RES	NV	PTR	PTH		
OnError					
ERR	symb	NULL			
Example					
Virtual System Path or User defined path are substituted (the real path +all the left side)					
_PWK_PKG_ Path of Package					
_PWK_REL_ Path relative to actual program path					
_PWK_STC_ Static web address					

23.6 WUI_SPTH

WUI_SPTH		fnc
Description	Set Path for Symbolic path translation	
Related	WUI_CPTH	
Remark		

1.WUI_SPTH					
Parameter	Type	Values	Comment	Default	Opt
WUI	PTR		Valid WUI ptr		
VPTH	symb		Virtual Path		
RPTH	symb		Real Path		
Return					
RES	NV	PTR	WUI		
OnError					
ERR	symb	NULL			
Example					
Predefined System Path: _PWK_PKG_ Path of Package _PWK_REL_ Path relative to actual program path _PWK_STC_ Static web address					



24

MTH

24.1 MTH_PLY

MTH_PLY		fnc
Description	Polynomial	
Related		
Remark	TBL should have X, Y column name	

1.MTH_PLY		Creation or recompute			
Parameter	Type	Values	Comment	Default	Opt
PLY	PTR		If null a new poly is created		
TBL	PTR		Table with X,Y values (if null) old values are used		
ORD	NV		POLYnomial Order (<10)		
XCOLNAM	symb		Name of X column	£X	YES
YCOLNAM	symb		Name of Y column	£Y	YES
EL_NUM	NV		Number of elements	TBL	YES
EL_FIRST	NV		Start element 1bsd	1	YES
Return					
RES	NV	PTR	PLY		
OnError					
ERR	symb	NULL			
Example					
<pre>ply=MTH_PLY(£NULL, tbl, 5) ; !! a ply is created and computed with order=5 MTH_PLY(ply, £NULL, 7) ; !! Recomputed with order=7 MTH_PLY(ply, tbl1, 6) ; !! new values computed with order=6</pre>					

2.MTH_PLY					
Parameter	Type	Values	Comment	Default	Opt
PLY	PTR		valid pTR		
WHAT	NNV	£X,£Y	Value to compute		
VAL	NV		y or x value		
Return					
RES	NV		X or Y computed value		
OnError					
ERR	symb	NULL			
Example					
<pre>x=MTH_PLY(PLY, £X, y) ; y=MTH_PLY(PLY, £Y, x) ;</pre>					



25

HIO

25.1 HIO_GPIOnew

HIO_GPIOnew					fnc
Description	Open a GPIO device (Default device)				
Related					
Remark					

1.HIO_GPIOnew					
Parameter	Type	Values	Comment	Default	Opt
Return					
RES	PTR		PTR to HIO GPIO		
OnError					
RES	symb	NULL			
Example					



25.2 HIO_GPIOset

HIO_GPIOset		fnc
Description	Set parameters of a GPIO device	
Related		
Remark	NULL value= not change; Empty (£) =reset	

1.HIO_GPIOset					
Parameter	Type	Values	Comment	Default	Opt
GPIO	PTR		A PTR to a GPIO device pin		
PIN	NV		The pin number		
MODE	symbol	0(input);1(output);2(inputPullUp);3(inputPullDown);4(outputOpenDrain);5(outputOpenDrainPullUp);6(outputOpenSource);7(outputOpenSourcePullDown)	The open mode, input		YES
TRIG	PTR		PTR to a trig		YES
TRIG_WAIT	NV		Isteresys time		YES
Return					
RES	NV		PTR to HIO GPIO		
OnError					
RES	symb	NULL			
Example					
The trig set these parameters: trg_pin = the pin number trg_val = the pin value trg_edge = the edge value					

25.3 HIO_GPIO

HIO_GPIO		fnc
Description	INPUT or Output on a pin	
Related		
Remark	if the pin is not configured, the first operation set it to with the default mode (0 or 1)	

1.HIO_GPIO					
Parameter	Type	Values	Comment	Default	Opt
GPIO	PTR		A PTR to a GPIO device pin		
PIN	NV		The pin number		
OUT_VALUE	symbol	0;£LOW(0);1;£HIGH(1)	if NUL an input operation is performed		YES
Return					
RES	NV		pin Value		
OnError					
RES	symb	NULL			
Example					



25.4 HIO_GPIOpwm

HIO_GPIOpwm		fnc
Description	Return a PTR to a PWM manager on a pin	
Related		
Remark		

1.HIO_GPIOpwm					
Parameter	Type	Values	Comment	Default	Opt
GPIO	PTR		A PTR to a GPIO		
PIN	NV		The pin number		
CYCLE	NV		in micro the cycle		YES
Return					
RES	PTR		PTR TO HIO_PWM		
OnError					
RES	symb	NULL			
Example					

25.5 HIO_GPIOpulse

HIO_GPIOpulse		fnc
Description	Raise a pulse on a pin	
Related		
Remark		

1.HIO_GPIOpulse					
Parameter	Type	Values	Comment	Default	Opt
GPIO	PTR		A PTR to a GPIO		
PIN	NV		The pin number		
VALUE	NV	1;0	in micro the cycle		
MICRO	NV		Duration in micro		
Return					
RES	PTR		TO GPIO		
OnError					
RES	symb	NULL			
Example					



25.6 HIO_PWM

HIO_PWM		fnc
Description	Operation on a PWM	
Related		
Remark		

1.HIO_PWM					
Parameter	Type	Values	Comment	Default	Opt
PWM	PTR		A PTR to a PWM		
CMD	symbol	£START;£STOP; £SET			
DUTY	NV		in micro the duty part of the cycle		YES
CICLE	NV		in micro the cicle		YES
Return					
RES	PTR		PTR TO HIO_PWM		
OnError					
RES	symb	NULL			
Example					

25.7 HIO_I2Cnew

HIO_I2Cnew		fnc
Description	Open an I2C device	
Related	SPI, UART	
Remark		

1.HIO_I2Cnew					
Parameter	Type	Values	Comment	Default	Opt
DEVICE	symbol	£I2C;£I2C1;£I2C2	The bus name		
UNIT_ADR	NV		The unit address		
UNIT_ID	symbol		The unit ID		YES
BUS_SPEED	symbol	0(Standard);1(fast)	Bust speed		YES
Return					
RES	PTR		PTR to HIO I2C device		
OnError					
RES	symb	NULL			
Example					
DEVICE are renamed according to board type: PI 2 /3 I2C1 DRAGONBOARD I2C0, I2C1 MINNOWBOARD I2C5					



25.8 HIO_I2C

HIO_I2C		fnc
Description	read and write on i2C device to and from a BUF or directly	
Related	BUF	
Remark		

1.HIO_I2C		R/W From buffer			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		I2C device PTR		
ACTION	symbol	£READ;£WRITE; £WriteRead	WriteRead makes a read with Restart instead of stop		
BUFFER	PTR		A BUF PTR		
BYTES	NV		Number of bytes to read or write		YES
REGISTER	NV		ONLY FOR £READ and £WriteRead Register Number (write to dvc before the read)		YES
Return					
RES	PTR		PTR to BUF		
OnError					
RES	symb	NULL			
Example					
<pre>buf=BUF_NEW(2,£U8); BUF_VAL(buf,1,0x5); BUF_VAL(buf,2,0x8); HIO_I2C(dvc,£WRITE, buf,1); !!write only 0x5; HIO_I2C(dvc,£WRITE, buf); !!write all the buffer 0x5, 0x8 ; HIO_I2C(dvc,£READ, buf,1); !! read 1 byte; HIO_I2C(dvc,£READ, buf); !! read all the buffer :2 bytes; HIO_I2C(dvc,£READ, buf,2,0xA); !! writes 0xA (Register)and read all the buffer :2 bytes (Write part Accept multiple bytes like W Direct) TRASH(buf);</pre>					

2.HIO_I2C		W Direct			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		I2C device PTR		
ACTION	symbol	£WRITE		£WRITE	
BUFFER	PTR	NULL		NULL	YES
BYTES	slis		symbol to write		YES
Return					
RES	PTR		PTR to BUF		
OnError					
RES	symb	NULL			
Example					
HIO_I2C(dvc,£WRITE, NULL,0x5,0x8); !!write 0x5, 0x8;					

3.HIO_I2C		R Directly			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		I2C device PTR		
ACTION	symbol	£READ		£READ	
BUFFER	PTR	NULL		NULL	YES
BYTES	NV		number of byte to read (1,2,3,4)	1	YES
REGISTER	NV		Register Number (written to dvc before the read)		YES
Return					
RES	symb		data read		
OnError					
RES	symb	NULL			
Example					
s=HIO_I2C(dvc,£READ, NULL,1); !!read 1 byte (maximum 4) ;					



25.9 HIO_SPInew

HIO_SPInew		fnc
Description	Open an SPI device	
Related	I2C, UART	
Remark		

1.HIO_SPInew					
Parameter	Type	Values	Comment	Default	Opt
DEVICE	symbol	£SPI;£SPI1;£SPI2	The bus name		
LINE	NV		Selection line		
UNIT_ID	symbol		The unit ID		YES
CLOCK_FRQ	NV		clock Frequency Hz		YES
BIT_Lenght	NV				YES
MODE	NV	0(CPOL 0 - CPHA 0); 1(CPOL 0 - CPHA 1); 2(CPOL 1 - CPHA 0); 3(CPOL 1 - CPHA 1);	Communication Mode, Clock Polarity and Clock Phase		
Return					
RES	PTR		PTR to HIO SPI device		
OnError					
RES	symb	NULL			
Example					
DEVICE are renamed according to board type: PI 2 /3 SPI0, SPI1 DRAGONBOARD SPI0 MINNOWBOARD SPI0					

25.10 HIO_SPI

HIO_SPI		fnc
Description	read and write on SPI device to and from a BUF or directly	
Related	BUF	
Remark		

1.HIO_SPI		R/W From buffer			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		SPI device PTR		
ACTION	symbol	£READ;£WRITE			
BUFFER	PTR		A BUF PTR		
BYTES	NV		Number of bytes to read or write		YES
REGISTER			ONLY FOR £READ Register Number (write to dvc before the read)		YES
Return					
RES	PTR		PTR to BUF		
OnError					
RES	symb	NULL			
Example					
<pre>buf=BUF_NEW(2,£U8); BUF_VAL(buf,1,0x5); BUF_VAL(buf,2,0x8); HIO_SPI(dvc,£WRITE, buf,1); !!write only 0x5; HIO_SPI(dvc,£WRITE, buf); !!write all the buffer 0x5, 0x8 ; HIO_SPI(dvc,£READ, buf,1); !! read 1 byte; HIO_SPI(dvc,£READ, buf); !! read all the buffer :2 bytes; TRASH(buf);</pre>					

2.HIO_SPI		W Directly			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		SPI device PTR		
ACTION	symbol	£WRITE		£WRITE	
BUFFER	PTR	NULL		NULL	YES
BYTES	slis		symbol to write		YES
Return					
RES	PTR		PTR to BUF		
OnError					
RES	symb	NULL			
Example					
HIO_SPI(dvc,£WRITE, NULL,0x5,0x8); !!write 0x5, 0x8;					



3.HIO_SPI		R Directly			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		SPI device PTR		
ACTION	symbol	£READ		£READ	
BUFFER	PTR	NULL		NULL	YES
BYTES	NV		number of byte to read (1,2,3,4)	1	YES
REGISTER	NV		Register Number (written to dvc before the read)		YES
Return					
RES	symb		data read		
OnError					
RES	symb	NULL			
Example					
s=HIO_SPI(dvc,£READ, NULL,1); !!read 1 byte (maximum 4) ;					

25.11 HIO_UARTnew

HIO_UARTnew		fnc
Description	Open an UART device	
Related	SPI, I2C	
Remark		

1.HIO_UARTnew					
Parameter	Type	Values	Comment	Default	Opt
DEVICE	symbol	£UART; £UART1;£UART1	The bus name		
UNIT_ID	symbol		The unit ID		YES
Return					
RES	PTR		PTR to HIO UART device		
OnError					
RES	symb	NULL			
Example					
DEVICE are renamed according to board type: PI 2 /3 UART0 DRAGONBOARD UART0,UART1 MINNOWBOARD UART1,UART2					



25.12 HIO_UARTset

HIO_UARTset		fnc
Description	Set communication parameter	
Related		
Remark		

1.HIO_UARTset		R/W From buffer			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		UART device PTR		
BAUD	NV		Baud rate		YES
DATABit	NV		number of data bit		YES
STOPbit	symb	£1;£1.5;£2	stop bit		YES
PARITY	symb	£NONE;£ODD; £EVEN;£MARK; £SPACE			YES
HANDshake	symb	£NONE;£RTS; £XonXoff; £RtsXonXof			YES
TXtmo	NV		TX tmo in ms		YES
RXtmo	NV		RX tmo in ms		YES
Return					
RES	PTR		PTR to BUF		
OnError					
RES	symb	NULL			
Example					

25.13 HIO_UART

HIO_UART		fnc
Description	read and write on UART device to and from a BUF or directly	
Related	BUF	
Remark		

1.HIO_UART		R/W From buffer			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		UART device PTR		
ACTION	symbol	£READ;£WRITE			
BUFFER	PTR		A BUF PTR		
BYTES	NV		Number of bytes to read or write		YES
Return					
RES	NV		Number of Bytes transferred		
OnError					
RES	symb	NULL			
Example					
<pre> buf=BUF_NEW(2,£U8); BUF_VAL(buf,1,0x5); BUF_VAL(buf,2,0x8); HIO_UART(dvc,£WRITE, buf,1); !!write only 0x5; HIO_UART(dvc,£WRITE, buf); !!write all the buffer 0x5, 0x8 ; HIO_UART(dvc,£READ, buf,1); !! read 1 byte; HIO_UART(dvc,£READ, buf); !! read all the buffer :2 bytes; TRASH(buf); </pre>					

2.HIO_UART		W Directly			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		UART device PTR		
ACTION	symbol	£WRITE		£WRITE	
BUFFER	PTR	NULL		NULL	YES
BYTES	slis		symbol to write		YES
Return					
RES	NV		Number of Bytes transferred		
OnError					
RES	symb	NULL			
Example					
HIO_I2C(dvc,£WRITE, NULL,0x5,0x8); !!write 0x5, 0x8;					



3.HIO_UART		R Directly			
Parameter	Type	Values	Comment	Default	Opt
DEVICE	PTR		UART device PTR		
ACTION	symbol	£READ		£READ	
BUFFER	PTR	NULL		NULL	YES
BYTES	NV		number of byte to read (1,2,3,4)	1	YES
Return					
RES	symb		data read		
OnError					
RES	symb	NULL			
Example					
s=HIO_I2C(dvc,£READ, NULL,1); !!read 1 byte (maximum 4) ;					

26 NPR



27

OCV

27.1 OCV_MAT

OCV_MAT		fnc
Description	Create or make action on a MAT	
Related		
Remark		

1.OCV_MAT		Create			
Parameter	Type	Values	Comment	Default	Opt
WHAT	symb	£NEW			
COL	NV		Number of column		
ROW	NV		Number of Row		
TYPE	symb		OCV data type		
BUF	PTR		Buffer with data		YES
Return					
RES	PTR		PTR to MAT		
OnError					
RES	symb	NULL			
Example					
mat=OCV_MAT(£NEW,800,600, £CV_8UC3); !! three channel U8 e.g. BGR; mat=OCV_MAT(£NEW,800,600, £CV_32FC1,buf); !! one channel FLT32 data from buffer ; OCV TYPE: [CV_] {8U,8S,16U,16S,32S,32F,64F}Cx Where x is the number of the channel example: CV_8UC1, 8UC3, 32FC1					

2.OCV_MAT		Update inner data			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to Mat		
Return					
RES	PTR		PTR to MAT		
OnError					
RES	symb	NULL			
Example					
Inner Mat data are used in function that are not part of OCV lib e.g. GUI_STREAM. In some situation inner Mat data can differ from real. Calling OCV_MAT(mat) assure that the data are in sync.					



3.OCV_MAT		Clone			
Parameter	Type	Values	Comment	Default	Opt
MAT	symb		PTR to MAT		
WHAT	symb	£CLN	clone		
Return					
RES	PTR		PTR to cloned MAT		
OnError					
RES	symb	NULL			
Example					
<pre>cln=OCV_MAT(mat,£CLN);</pre>					

4.OCV_MAT		Copy To			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to MAT		
WHAT	symb	£CPYTO	Copy To		
DEST	PTR		If not provided is created		YES
MASK	PTR		Mat mask CV_U8C1		YES
Return					
RES	PTR		PTR to dest MAT		
OnError					
RES	symb	NULL			
Example					
<pre>dst=OCV_MAT(mat,£CPYTO, dst, msk);</pre> <p>mat point are copied in dst for values of msk >0; mat,dst,msk should have the same dimensions.</p>					

5.OCV_MAT		Convert Mat element type in a NEW mat			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to MAT		
WHAT	symb	£CNV	Convert Element type		
TYPE	symb		ocv element type like CV_8U		
ALPHA	FLT		see OpenCv documentation		YES
BETA	FLT				YES
Return					
RES	PTR		PTR to new MAT		
OnError					
RES	symb	NULL			
Example					
<pre>mat=OCV_MAT(£NEW,800,600, £CV_32FC3); nmt=OCV_MAT(mat,£CNV,£CV_32S);</pre>					

6.OCV_MAT		Convert Mat color in a NEW mat			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to MAT		
WHAT	symb	£CNVCLR	Convert mat Color space		
COLOR_CODE	symb		see OpenCv documentation		
Return					
RES	PTR		PTR to new MAT		
OnError					
RES	symb	NULL			
Example					
<pre>nmt=OCV_MAT(mat,£CNVCLR,£COLOR_BGR2GRAY); nmt=OCV_MAT(mat,£CNVCLR,6); !!same conversion but using the code number;</pre>					

7.OCV_MAT		Return a PWK BUF of the RAW data			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to MAT		
WHAT	symb	£GETBUF	Convert mat Color space		
Return					
RES	PTR		PTR to BUF		
OnError					
RES	symb	NULL			
Example					
<pre>buf=OCV_MAT(mat, £GETBUF);</pre>					



27.2 OCV_MAT_INF

OCV_MAT_INF		fnc
Description	Return MAT info	
Related		
Remark		

1.OCV_MAT_INF		Single info element			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to MAT		
WHAT	symb	£COL;£ROW; £DATSIZ;£TOTELM; £CHN(channel num);£CHNtyp; £CHNsiz; £ELMtyp(element typ); £ELMsiz;FRMT(for mat string)			
Return					
RES	symb		the request info		
OnError					
RES	symb	NULL			
Example					
ColNum=OCV_MAT_INF(mat,£COL);					
Format string is like : WIDTH=%d, heigh=%d, line=%d, pxf=%s					

2.OCV_MAT_INF		Get a TBL info			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to MAT		
WHAT	symb	£INFOTBL			
Return					
TBL	PTR		PTR to a TBL with one row of info		
OnError					
RES	symb	NULL			
Example					
tbl=OCV_MAT_INF(mat,£INFOTBL); colNum=TBL_ITM(tbl, £COL,1);					
Columns have the names used in single info use.					

27.3 OCV_MAT_PIX

OCV_MAT_PIX		fnc
Description	Set or Get Pixel	
Related		
Remark		

1.OCV_MAT_PIX		Set Get Pixel value			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to MAT		
X	NV		Column		
Y	NV		Row		
CHN	symb		sequence of channel num		
VALUE	symb		in Hex or as "v1;v2;vx"		
RetFRMT	Symb	NULL;£HEX			OPT
Z	NV				OPT
Return					
RES	symb		the VALUE		
OnError					
RES	symb	NULL			
Example					
<pre> res=OCV_MAT_PIX(mat,10,20,"123","64;128;0"); !!set values: ch1=64, ch2=128, ch3=0; res=OCV_MAT_PIX(mat,10,20,"123",0xFF8000); !!set values: ch1=255, ch2=128, ch3=0; res=OCV_MAT_PIX(mat,10,20,"123"); !!return: "255;128;0"; res=OCV_MAT_PIX(mat,10,20,"321"); !!return: "0;128;255"; res=OCV_MAT_PIX(mat,10,20,"123",NULL,£HEX); !!return: 0xFF8000; </pre>					



27.4 OCV_OP

OCV_OP		fnc
Description	Operation with mat	
Related		
Remark		

1.OCV_OP		Operation with mat			
Parameter	Type	Values	Comment	Default	Opt
First	PTR				
OPERATOR	symb	£SUM;£SUB;£MUL; £DIV;£>;£GT;£>; £GE;£<;£LT;£<; £LE;£AND;£OR; £XOR;£MIN;£MAX; £CROSS;£DOT; £ABS;£INV	Frame to add		
Second	PTR/NV		In some operation you can use NV		YES
Return					
MAT	PTR		Result in a new mat		
OnError					
RES	symb	NULL			
Example					
matRes=MAT_OP(mat1; £SUM; mat2);					
Operation allowed also with NV: MUL, DIV, GT, GE, LT, LE, MIN, MAX.					
Operation with single operator: ABS, INV					

27.5 OCV_BKGSUB

OCV_BKGSUB		fnc
Description	BackGroud Progrssive Subtractor	
Related		
Remark		

1.OCV_BKGSUB		Create Subtractor			
Parameter	Type	Values	Comment	Default	Opt
BkgMAT	PTR	£NULL			OPT
BKGsub	PTR	£NULL			OPT
RetType	symb	£PTR			OPT
LearnRate	FLT		Value from -1.0 to 1.0	-1.0	OPT
TYPEofSub	symb	£MOG2;£kNN		£MOG	OPT
SENSIBILITY	FLT		Default MOG2=16.0 Knn=400		OPT
SHADOW	symb	£FALSE; £TRUE		£FALSE	OPT
Return					
BKS	PTR		PTR of SUBtractor		
OnError					
RES	symb	NULL			
Example					
If BkgMat is provided is used as first background to learn. Learn rate : -1.0 = automatic, 0=not update, 1=reinit – value between 0> <1 indicates how fast the model is learnt. TYPEofSUB, SENSIBILITY, SHADOW take effects only at creation time.					



2.OCV_BKGSUB		Update the model			
Parameter	Type	Values	Comment	Default	Opt
BkgMAT	PTR				
BKGsub	PTR				
RetType	symb	£PTR;£MSK;£FRG		£PTR	OPT
LearnRate	FLT		Value from -1.0 to 1.0	-1.0	OPT
BKS or FRG or MSK	PTR		depending from retType		
OnError					
RES	symb	NULL			
Example					
If BkgMat is provided is used as first background to learn. Learn rate : -1.0 = automatic, 0=not update, 1=reinit – value between 0> <1 indicates how fast the model is learnt. TYPEofSUB, SENSIBILITY, SHADOW take effects only at creation time.					

27.6 OCV_BKGRMV

OCV_BKGRMV		fnc
Description	Remove the background from a mat	
Related		
Remark		

1.OCV_BKGRMV		Remove the background			
Parameter	Type	Values	Comment	Default	Opt
FRGMAT	PTR		Foreground		
BKG	PTR		Background MAT or BKGSUB		
RetType	symb	£MSK;£FRG		£MSK	OPT
TYPEofSub	symb	£MOG2;£kNN		£MOG	OPT
SENSIBILITY	FLT		Default MOG2=16.0 Knn=400		OPT
SHADOW	symb	£FALSE; £TRUE		£FALSE	OPT
NFRG	PTR				OPT
Return					
FRG or MSK	PTR		depending of RetType		
OnError					
RES	symb	NULL			
Example					
<p>if BKG is a Background Subtractor is is used and TYPEofSub, SENSIBILITY, SHADOW hve no effect. If RetType is £FRG and NFRG is provided the obtained foreground is copied into else a the result is put in a new mat as for £MSK</p>					



27.7 OCV_IMREAD

OCV_IMREAD		fnc
Description	Read an image file	
Related		
Remark	see OpenCv documentation fro detail	

1.OCV_IMREAD		Read an image into a mat			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR				
FILENAME	symb				
FLAG	symb	£UNCHANGED; £GRAYSCALE; £COLOR ; £ANYDEPTH; £ANYCOLOR; £LOAD_GDAL; £REDUCED_GRAYSCALE_2; £REDUCED_COLOR_2; ; £REDUCED_GRAYSCALE_4; £REDUCED_COLOR_4; ; £REDUCED_GRAYSCALE_8; £REDUCED_COLOR_8; ; £IGNORE_ORIENTATION = 128		UNCHANGED	OPT
Return					
MAT	PTR				
OnError					
RES	symb	NULL			
Example					
UNCHANGED = -1, //!< If set, return the loaded image as is (with alpha channel, otherwise it gets cropped). GRAYSCALE = 0, //!< If set, always convert image to the single channel grayscale image. COLOR = 1, //!< If set, always convert image to the 3 channel BGR color image. ANYDEPTH = 2, //!< If set, return 16-bit/32-bit image when the input has the corresponding depth, otherwise convert it to 8-bit. ANYCOLOR = 4, //!< If set, the image is read in any possible color format. LOAD_GDAL = 8, //!< If set, use the gdal driver for loading the image. REDUCED_GRAYSCALE_2 = 16, //!< If set, always convert image to the single channel grayscale image and the image size reduced 1/2. REDUCED_COLOR_2 = 17, //!< If set, always convert image to the 3 channel BGR color image and the image size reduced 1/2. REDUCED_GRAYSCALE_4 = 32, //!< If set, always convert image to the single channel grayscale image and the image size reduced 1/4. REDUCED_COLOR_4 = 33, //!< If set, always convert image to the 3 channel BGR color image and the image size reduced 1/4. REDUCED_GRAYSCALE_8 = 64, //!< If set, always convert image to the single channel grayscale image and the image size reduced 1/8. REDUCED_COLOR_8 = 65, //!< If set, always convert image to the 3 channel BGR color image and the image size reduced 1/8. IGNORE_ORIENTATION = 128 //!< If set, do not rotate the image according to EXIF's orientation flag.					

27.8 OCV_IMWRITE

OCV_IMWRITE		fnc
Description	Write an Image on a file	
Related	OCV_MAT:CNV, OCV_MAT:CNVCLR	
Remark	The image format is chosen based on the filename extension see OpenCv documentation for detail	

1.OCV_IMWRITE		Save a mat on a file			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		PTR to MAT		
FILENAME	symb		The file extension is		
FLAG	symb	£BGR2RGB; £BGR2RGBA			
Return					
RES	symb		the MAT		
OnError					
RES	symb	NULL			
Example					
<p>In general, only 8-bit single-channel or 3-channel (with 'BGR' channel order) images can be saved using this function, with these exceptions:</p> <p>16-bit unsigned (CV_16U) images can be saved in the case of PNG, JPEG 2000, and TIFF formats 32-bit float (CV_32F) images can be saved in PFM, TIFF, OpenEXR, and Radiance HDR formats; 3-channel (CV_32FC3) TIFF images will be saved using the LogLuv high dynamic range encoding (4 bytes per pixel) PNG images with an alpha channel can be saved using this function. To do this, create 8-bit (or 16-bit) 4-channel image BGRA, where the alpha channel goes last. Fully transparent pixels should have alpha set to 0, fully opaque pixels should have alpha set to 255/65535 (see the code sample below). If the format, depth or channel order is different, use OCV_Mat::VNV and OCV_Mat:CNVCLR to convert it before saving.</p> <p>FILE EXTENSION</p> <p>Windows bitmaps - *.bmp, *.dib (always supported) JPEG files - *.jpeg, *.jpg, *.jpe (see the Note section) JPEG 2000 files - *.jp2 (see the Note section) Portable Network Graphics - *.png (see the Note section) WebP - *.webp (see the Note section) Portable image format - *.pbm, *.pgm, *.ppm *.pxm, *.pnm (always supported) PFM files - *.pfm (see the Note section) Sun rasters - *.sr, *.ras (always supported) TIFF files - *.tiff, *.tif (see the Note section) OpenEXR Image files - *.exr (see the Note section) Radiance HDR - *.hdr, *.pic (always supported) Raster and Vector geospatial data supported by GDAL (see the Note section)</p>					



27.9 OCV_VCP

OCV_VCP		fnc
Description	Create an ocv Video Capture	
Related	OCV_VCP_RD	
Remark		

1.OCV_VCP		Create an ocv Video Capture			
Parameter	Type	Values	Comment	Default	Opt
Source	symb				
Return					
OCV_VCP	PTR		PTR to vcp		
OnError					
RES	symb	NULL			
Example					
vcp=OCV_VCP(0); !!create on camera 0; ocv_vcp("rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny_115k.mov")					

27.10 OCV_VCP_RD

OCV_VCP_RD		fnc
Description	Read a MAT from VCP	
Related	OCV_VCP	
Remark		

1.OCV_VCP_RD		Reat a frame			
Parameter	Type	Values	Comment	Default	Opt
OCV_VCP	PTR				
Return					
MAT	PTR		The frame		
OnError					
RES	symb	NULL			
Example					
mat=OCV_VCP_RD(vcp);					



27.11 OCV_VWR

OCV_VWR		fnc
Description	Create a video writer	
Related	OCV_VWR_WR	
Remark		

1.OCV_VWR		Create a VWR			
Parameter	Type	Values	Comment	Default	Opt
FILENAME	symb				
FCC	symb	£MJPG;£PIM1	Encoder FCC		
FPS	NV		Frame per second		
SX	NV		Size X		
SY	NV		Size Y		
Return					
VWR	PTR		Pointer to Video Writer		
OnError					
RES	symb	NULL			
Example					
vwr=OCV_VWR("MyMovie.mjpg", £MKPG, 30, 800,600);					

27.12 OCV_VWR_WR

OCV_VWR_WR		fnc
Description	Write a frame	
Related	OCV_VWR	
Remark		

1.OCV_VWR_WR		write a frame			
Parameter	Type	Values	Comment	Default	Opt
VWR	PTR				
MAT	PTR		Frame to add		
Return					
VWR	PTR		Pointer to Video Writer		
OnError					
RES	symb	NULL			
Example					
mat=OCV_VCP_WR(vwr, mat);					



27.13 OCV_QR

OCV_QR		fnc
Description	Read a QR code	
Related		
Remark		

1.OCV_QR		Decode QRcode			
Parameter	Type	Values	Comment	Default	Opt
MAT	PTR		The image		
WHT	symb	£DETECTDECODE; £ZBAR			£ZBAR
Return					
RES	symb/PTR		DEcoded TEXT or if Zbar TBL of result		
OnError					
RES	symb	NULL			
Example					
text=OCV_QR(mat, £DETECTDECODE); TBL=OCV_QR(mat); the TBL contains the decoded texts and their coordinates Columns are TYPE,TEXT, X1,Y1 X2,Y2, X3,Y3 X4,Y4					

27.14 OCV_RS2

OCV_RS2		fnc
Description	Operation on a Real sense Camera create, Start/Stop	
Related	OCV_RS2_GET, OCV_VCP	
Remark		

1.OCV_RS2		Open a RS2 camera			
Parameter	Type	Values	Comment	Default	Opt
Return					
RES	PTR		PTR To RS		
OnError					
RES	symb	NULL			
Example					
rs2=OCV_RS2;					

2.OCV_RS2		START, STOP, READ			
Parameter	Type	Values	Comment	Default	Opt
RS2	PTR				
WHAT	symb	£READ; £STOP;START			
ALIGN	symb	£ALGN_DPT; £ALGN_CLR	align size to deep or color		YES
Return					
RES	PTR/NV		Read return a PTR to RS2 Frame or 0 if no frame, start ret 1, stop ret 1,		
OnError					
RES	symb	NULL			
Example					
if not start a read operation make an implicit start.					
rs2=OCV_RS2; frm=OCV_RS2(rs2,£READ, £ALGN_DPT);					



27.15 OCV_RS2_GET

OCV_RS2_GET		fnc
Description	Retrive data from a RS2 frame	
Related	OCV_RS2	
Remark		

1.OCV_RS2_GET		Get Frame			
Parameter	Type	Values	Comment	Default	Opt
RS2_FRAME	PTR		Obtained from OCV_RS2:READ		
WHAT	symb	£IFR(infrared frame);£CLR(color frame);£DPT(deep frame)		£CLR	OPT
DPT_PAR	symb	£NULL(distance); £CLRZ(Colorize); £MTR(mat of distance in meter)		£NULL	OPT
Return					
MAT	PTR				
OnError					
RES	symb	NULL			
Example					
rs2=OCV_RS2; frm=OCV_RS2(rs2,£READ, £ALGN_DPT); mat=OCV_RS2_GET(frm,£DPT,£MTR);					

2.OCV_RS2_GET		Get point distance between two point on the frame			
Parameter	Type	Values	Comment	Default	Opt
RS2_FRAME	PTR		Obtained from OCV_RS2:READ		
WHAT	symb	£MSR	Measure		
X1	NV		Point 1		
Y1	NV				
X2	NV		Point 2		
Y2	NV				
Return					
RES	PTR		distance in meter		
OnError					
RES	symb	NULL			
Example					
dist=OCV_RS2_GET(frm,£MSR, 10,10,200,200);					

28

OPC

This lib is a Wrapper for the OPEN62541 (open62541.org) that is a GREAT implementation of the OPC Standard (many thanks to the dev Group).

But .. using OPC is like shooting a fly with a cannon ..

In our opinion OPC is a “*cervellotic*” project, full of unnecessary complications and many design ingenuities.

But .. it is a Standard ..

We have done our best to make things simple but anyway, apart from simple case like read or write a simple data, you have to study the OPC documentation.



28.1 OPC_NEW

OPC_NEW		fnc
Description	Create an OPC server or Client	
Related		
Remark		

1.OPC_NEW					
Parameter	Type	Values	Comment	Default	Opt
WHAT	symb	£CLI;£SRV			
PAR	symb		For SRV the TCP port		YES
Return					
RES	PTR		PTR to OPC		
OnError					
RES	symb	NULL			
Example					

28.2 OPC_SRV

OPC_SRV		fnc
Description	Start or Stop an OPC SRV	
Related		
Remark		

1.OPC_SRV					
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
CMD	symb	£START;£STOP			
Return					
RES	PTR		PTR to OPC		
OnError					
RES	symb	NULL			
Example					

28.3 OPC_CLI

OPC_CLI		fnc
Description	Open, Close or Set Parameters for an OPC Client connection	
Related		
Remark		

1.OPC_CLI		Open / Close			
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
CMD	symb	£OPEN;£CLOSE			
EPS	symb		End Point String for OPEN		YES
Return					
RES	PTR		PTR to OPC		
OnError					
RES	symb	NULL			
Example					
opc_ptr=OPC_CLI(opc_ptr, £OPEN, "opc.tcp://localhost:4840");					

2.OPC_CLI		Set Parameters			
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
CMD	symb	£SETPAR			
TRIG	symb		PTR to connection Trig		YES
TMITER	NV		Iteration time (ms)		YES
TMITERSleep	NV		Sleep Time between iteration (ms)		YES
Return					
RES	PTR		PTR to OPC		
OnError					
RES	symb	NULL			
Example					
<p>TRIG parameters are : CLI_STATUS, OPC_PTR</p> <p>CLI_STATUS: DISCONNECTED, WAIT_ACK, TCP_CONNECT, SECURED, CONNECTED, SESSION_DISCONNECTED, SESSION_RENEWED</p> <p>In PWK OPC implementation a thread takes care of asynchronous communication, the iteration time is the Working time for communications with the Server, the ltersleep is the pause between WT.</p>					



28.4 OPC_RD

OPC_RD		fnc
Description	Read	
Related	OPC_GET, OPC_WR	
Remark	This function return a PTR (ToDel) that contains the read value. This because can return a plain symbol or a TBL or a BUF	

1.OPC_RD					
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
WHAT	symb	£VALUE; £DISPLAYNAME; £DESCRIPTION; £INVERSENAME; £BROWSENAME; £DATATYPE; £NODEID; £BROWSE; £SUBID(Subscriptio l d)			
NODEID	symb		Name space + Name id "n;id" or "ns:n ; s=id"		YES
AS	symb	£VAL;£TBL;£BUF		£VAL	YES
Return					
SUBID	NV		If WHAT=SUBID		
RES	PTR		PTR to XUA VAL to read use OPC_GET		
OnError					
RES	symb	NULL			
Example					
rs=OPC_RD(opc_ptr,£VAL,"2;.recipe"); recipe=OPC_GET(rs);					

28.5 OPC_GET

OPC_GET		fnc
Description	Get the value or the kind, or the type of a result of an OPC function	
Related	OPC_RD	
Remark		

1.OPC_GET					
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
WHAT	symb	£VAL;£KND(kind); £TYP	KND could be VAL,TBL,BUF TYP is the type (BO,U8 ..)	£VAL	YES
Return					
KND	symb		if(KND) could be VAL, TBL, BUF		
TYP	symb		if(TYP) BO,U8,I8,U16,I16,U32,I32,I64,U64,F32,F64,STR,DTT,GID		
VAL	symb		if(VAL) could be a plain symb or a PTR to TBL or BUF		
OnError					
RES	symb	NULL			
Example					
<pre>rs=OPC_RD(opc_ptr,£VAL,"2;.recipe"); #while(1); !!like a switch; #if(OPC_GET(rs,£KND)==£val); recipe=OPC_GET(rs); #break;#end; #if(OPC_GET(rs, £KND)==£TBL); recTbl=OPC_GET(rs); !! use recTbl then; TRASH(recTbl); #break;#end; #if(OPC_GET(rs, £KND)==£BUF); recBuf=OPC_GET(rs); !! use recBuf then; TRASH(recBuf); #break;#end; #break; #end; TRASH(rs); !! NOTE: if you don't read £val, you only need to trash the result NOT the contained VAL; DTT=DateTime (UNIX time)</pre>					



28.6 OPC_WR

OPC_WR		fnc
Description	Write	
Related	OPC_RD	
Remark		

1.OPC_WR					
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
NODEID	symb		Name space + Name id "n;id" or "ns:n ; s=id"		
VALUE	symb				
WHAT	symb	£VALUE			£VALUE
KND	symb	£VAL;£TBL;£BUF		£VAL	YES
TYP	symb	£BO; £U8;£I8;£U16;£I16; £U32;£I32;£I64;£U64; £F32;£F64;£STR; £DTT;£GID	If N.D. it is read before the write		YES
Return					
RES	NV		1=OK; 0=KO		
OnError					
RES	symb	NULL			
Example					
rs=OPC_WD(opc_ptr,"2;.recipe","MyRecipe");					

28.7 OPC_MNT

OPC_MNT		fnc
Description	Monitor an OPC Item for ValueAttribute change	
Related	OPC_RD	
Remark		

1.OPC_MNT					
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
NODEID	symb		Name space + Name id "n;id" or "ns:n ; s=id"		
TRIG	PTR		TRIG for data change		
Return					
RES	NV		SubID Subscription ID		
OnError					
RES	symb	NULL			
Example					
<pre>trg=TRIG("OPC\TRG_MNT","OPC_PTR, XUA_VAL,STATUS,TS_SERVER,TS_SOURCE"); rs=OPC_MNT(opc_ptr,"2;.recipe","MyRecipe",trg);</pre> <p>The XUA_VAL and the inner TBL or BUF PTR should be NEVER deleted because are deleted by the trig on return, if you want to use it in other side of the program you have to clone.</p>					



28.8 OPC_EVT

OPC_EVT		fnc
Description	Register for OPC events or Refresh	
Related	OPC_MNT	
Remark		

1.OPC_EVT		Set Event			
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
NODEID	symb		Name space + Name id "n;id" or "ns:n ; s=id"		
ATTID	NV		Attribute id for the event		
TRIG	PTR		PTR to TRIG for event		
CND	PTR		PTR to Condition Table		
Return					
RES	NV		>0 = OK (SUBID)		
OnError					
RES	symb	NULL			
Example					
<pre>trg_ev=TRIG(&TRIG_EV,"TBL_XUA_VAL"); tbl=TBL_NEW(NULL,3,NULL,NULL,"TypDefId;AttId;BrwPth"); tbl_itm(tbl,1,1,"0;2782"); !!! UA_NS0ID_CONDITIONTYPE; tbl_itm(tbl,2,1,"1"); !!! UA_ATTRIBUTEID_NODEID; tbl_itm(tbl,1,2,"0;2041"); !!! UA_NS0ID_BASEEVENTTYPE; tbl_itm(tbl,2,2,"13"); !!! UA_ATTRIBUTEID_VALUE; tbl_itm(tbl,3,2,"0;Message"); !!! UA_TYPES_QUALIFIEDNAME; tbl_itm(tbl,1,3,"0;2041"); !!! UA_NS0ID_BASEEVENTTYPE; tbl_itm(tbl,2,3,"13"); !!! UA_ATTRIBUTEID_VALUE; tbl_itm(tbl,3,3,"15;HelpText"); !!! UA_TYPES_QUALIFIEDNAME; ; opc_evt(opc,"ns=16;s=DiagnosisLogbook",12,trg_ev,tbl); !!12 is the attId for UA_ATTRIBUTEID_EVENTNOTIFIER;</pre>					
The trig is called with a TBL of XUA_VAL (one for each condition) organized for column and, as for MNT, don't delete any of this elements.					

2.OPC_EVT		Refresh event			
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
CMD	symb	£REFRESH			
Return					
RES	NV		SubID Subscription ID		
OnError					
RES	symb	NULL	PTR to OPC		
Example					
rs=OPC_EVT(opc_ptr,"£REFRESH);					



28.9 OPC_CALL

OPC_CALL		fnc
Description	Call an OPC Method	
Related	OPC_RD	
Remark		

1.OPC_CALL					
Parameter	Type	Values	Comment	Default	Opt
OPC	symb		OPC PTR		
OBJID	symb		Object id		
MTHDID	symb		Method id		
INPUTPAR	PTR		TBL of par COL(par, parTyp)		YES
Return					
RES	NV		TBL_XUA_VAL		
OnError					
RES	symb	NULL			
Example					
<pre>tbl=tbl_new(2,1); tbl_itm(tbl,1,0, OPC_RD(opc_ptr,£SUBID)); tbl_itm(tbl,2,0, £U32); !!UA_NS0ID_CONDITIONTYPE = 2782; !!UA_NS0ID_CONDITIONTYPE_CONDITIONREFRESH = 3875; tbl=OPC_CALL(opc_ptr, "0;2782","0; 3875", tbl); i=0; in=TBL_INF(tbl,£ROW); #while(i=i+1; i<in); TRASH(tbl_itm(tbl,1,i); #end; TRASH(tbl);</pre>					

28.10 OPC_DISCOVER

OPC_DISCOVER		fnc
Description	Discover end point	
Related	OPC_RD	
Remark		

1.OPC_DISCOVER					
Parameter	Type	Values	Comment	Default	Opt
WHAT	symb	£ENDPOINT			
QS	symb		Query String		
Return					
RES	NV		TBL_XUA_VAL		
OnError					
RES	symb	NULL			
Example					
<pre>res=OPC_DISCOVER(£Endpoint, "opc.tcp://192.168.2.17:4840"); chatput(£DISCOVER..tbl_exp(res)); TRASH(res)</pre>					



29

COM

Serial communication are done by the means of COM lib.

29.1 COM_NEW

COM_NEW		fnc
Description	Open a Serial COM	
Related		
Remark		

1.COM_NEW					
Parameter	Type	Values	Comment	Default	Opt
COM_NAME	symb		COM1, COM2, ..		
PARAMETERS	symb		See below for a list		
Return					
RES	PTR		PTR to COM		
OnError					
RES	symb	NULL	Error		
Example					
<p>com=COM_NEW(£COM3, "baud=9600 data=8 parity=n stop=2)</p> <p>[baud={11 110 15 150 30 300 60 600 12 1200 24 2400 48 4800 96 9600 19 19200}]</p> <p>[parity={n e o m s}] "none", "even", "odd", "mark", and "space".</p> <p>[data={5 6 7 8}]</p> <p>[stop={1 1.5 2}]</p> <p>[to={on off}] Specifies whether infinite time-out processing is on or off. The default is off.</p> <p>[xon={on off}] Specifies whether the xon or xoff protocol for data-flow control is on or off.</p> <p>[odsr={on off}] Specifies whether output handshaking that uses the Data Set Ready (DSR) circuit is on or off.</p> <p>[octs={on off}] Specifies whether output handshaking that uses the Clear To Send (CTS) circuit is on or off.</p> <p>[dtr={on off hs}] Specifies whether the Data Terminal Ready (DTR) circuit is on or off or set to handshake.</p> <p>[rts={on off hs tg}] Specifies whether the Request To Send (RTS) circuit is set to on, off, handshake, or toggle.</p> <p>[idsr={on off}] Specifies whether the DSR circuit sensitivity is on or off.</p> <p>The "baud" substring can be any of the values listed, which are in pairs. The two-digit values are the first two digits of the associated values that they represent.</p> <p>For example, 11 represents 110 baud, 19 represents 19,200 baud.</p> <p>The "parity" substring indicates how the parity bit is used to detect transmission errors.</p>					



29.2 COM_SET

COM_SET		fnc
Description	Change COM setting as well default timeOut values	
Related	COM_NEW	
Remark	Usually there is not need to change	

1.COM_SET					
Parameter	Type	Values	Comment	Default	Opt
COM	PTR				
PARAMETERS	symb		See CON_NEW		YES
RIT	NV		50 initial values		YES
RTTC	NV		50		YES
RTTM	NV		10		YES
WTTC	NV		50		YES
WTTM	NV		10		YES
Return					
RES	PTR		PTR to COM		
OnError					
RES	symb	NULL	Error		
Example					
<p>All values are in milliseconds.</p> <p>RIT: ReadIntervalTimeout Specifies the maximum time interval between arrival of two bytes. If the arrival time exceeds these limits the ReadFile() function returns.</p> <p>RTTC: ReadTotalTimeoutConstant is used to calculate the total time-out period for read operations. For each read operation, this value is added to the product of the ReadTotalTimeoutMultiplier member and the requested number of bytes.</p> <p>RTTM: ReadTotalTimeoutMultiplier is used to calculate the total time-out period for read operations. For each read operation, this value is multiplied by the requested number of bytes to be read.</p> <p>WTTC: WriteTotalTimeoutConstant similar to ReadTotalTimeoutConstant but for write operation.</p> <p>WTTM: WriteTotalTimeoutMultiplier similar to ReadTotalTimeoutMultiplier but for write operation.</p>					

29.3 COM_RCV

COM_RCV		fnc
Description	Receive data in a buffer	
Related	COM_WAIT	
Remark		

1.COM_RCV					
Parameter	Type	Values	Comment	Default	Opt
COM	PTR				
BUFFER	PTR				
BUFFER_INDEX	NV		Start of the buffer 1bsd	1	YES
SIZE	NV		# of byte to wait	0	YES
TMO	NV		Ms time out		YES
Return					
RES	NV		# of byte received		
OnError					
RES	symb	NULL	Error		
RES	NV		-1 Not valid COM setup -2=error in communication		
Example					
This function return when: <ul style="list-style-type: none"> - if SIZE when all byte are arrived, - when the buffer is full, - when TMO is expired. 					



29.4 COM_WAIT

COM_WAIT		fnc
Description	Wait for data	
Related	COM_WAIT	
Remark		

1.COM_WAIT					
Parameter	Type	Values	Comment	Default	Opt
COM	PTR				
BUFFER_OR_STOP	PTR		Use £STOP to end		
BUFFER_INDEX	NV		Start of the buffer 1bsd	1	YES
TMO	NV		Ms time out used after the first byte		YES
SIZE	NV		# of byte to wait		YES
Return					
RES	NV		# of byte received		
OnError					
RES	symb	NULL	Error		
RES	NV		-1 Not valid COM setup -2=error in communication		
Example					
This function return when: <ul style="list-style-type: none">- when the buffer is full or size is reached,- when TMO is expired, this time out start after the first received byte.- when in another THREAD the function is called with £STOP-					

29.5 COM_SND

COM_SND		fnc
Description	SEND data	
Related	COM_WAIT	
Remark		

1.COM_SND					
Parameter	Type	Values	Comment	Default	Opt
COM	PTR		PTR		
BUFFER	PTR		PTR		
BUFFER_INDEX	NV		1bsd		YES
SIZE	NV		In BYTE if ND buffer size is used		YES
Return					
RES	NV		# of byte sent		
OnError					
RES	symb	NULL	Error		
RES	NV		-1 Not valid COM setup -2=error in communication		
Example					



29.6 COM_FNC

COM_FNC		fnc
Description	Execute Escape Com function	
Related		
Remark		

1.COM_FNC					
Parameter	Type	Values	Comment	Default	Opt
COM	PTR		PTR		
FUNCTION	symb	£SETXOFF; £SETXON; £SETRTS; £CLRRTS; £SETDTR; £CLRDTR; £SETBREAK; £CLRBREAK			
Return					
RES	NV	1			
OnError					
RES	symb	NULL	Error		
RES	NV		-1 Not valid COM setup -2=error in communication		
Example					

29.7 COM_GMS

COM_GMS		fnc
Description	Get Modem status	
Related		
Remark		

1.COM_GMS					
Parameter	Type	Values	Comment	Default	Opt
COM	PTR		PTR		
WHAT	Symb	£CTS;£DSR; £RING;£CD			YES
Return					
RES	NV	1/0	If specified WHAT the status of		
RES	symb		The enum (comma separated) of the active signals		
OnError					
RES	symb	NULL	Error		
RES	NV		-1 Not valid COM setup -2=error in communication		
Example					
COM_GMS(P,£CTS) >>> 1/0 COM_GMS(P) >>> “CTS,CD”					



29.8 COM_ERR

COM_ERR		fnc
Description	Get Errors	
Related		
Remark		

1.COM_ERR					
Parameter	Type	Values	Comment	Default	Opt
COM	PTR		PTR		
WHAT	Symb	£DND;£IOE; £OOP;£PTO; £MODE; £BREAK; £FRAME; £RXOVER; £TXFULL; £OVERRUN;RXP ARITY			YES
Return					
RES	NV	1/0	If specified WHAT the status of		
RES	symb		The enum (comma separated) of the active errors		
OnError					
RES	symb	NULL	Error		
RES	NV		-1 Not valid COM setup -2=error in communication		
Example					

29.9 COM_STS

COM_STS		fnc
Description	Get COM status	
Related		
Remark		

1.COM_STS					
Parameter	Type	Values	Comment	Default	Opt
COM	PTR		PTR		
WHAT	Symb	£SNDQ(#char to send); £RCVQ(#char to read); £CTS_hold; £DSR_hold; £CD_hold;XOFF_hold;XOFF_sent;£EOF;£TXIM			YES
Return					
RES	NV	1/0	If specified WHAT the status of		
RES	symb		The enum (comma separated) of the active signals		
OnError					
RES	symb	NULL	Error		
RES	NV		-1 Not valid COM setup -2=error in communication		
Example					



List of Figures and Tables

Tabella 1.1: Esempio Tabella.....	9
-----------------------------------	---